The ISP Column
An occasional column on things Internet

September 2006

Geoff Huston

DNSSEC - The Practice

Last month's column looked at the theory of DNSSEC, examining the additional Resource Records that are defined by DNSSEC, how these Resource Records are interpreted, and the manner in which these additional RR's can be used to authenticate DNS responses. All this is fine in theory, but how well does all this work in practice? In this column I'd like to describe my experiences in configuring a DNSSEC-aware DNS resolver, and setting up a DNSSEC-secured zone, and see how easy, or hard, it is to use DNSSEC in practice.

The Objective

In this exercise I would like to use DNSSEC to sign two DNS zones.. The first zone, *dnssec.potaroo.net*, is to be a signed zone which is a descendant from the unsigned zone *potaroo.net*, The second zone, *sub.dnssec.potaroo.net*, is to be a signed immediate descendant zone from the first signed zone. In theory the one trust anchor should be sufficient to validate both zones. I also would like to check that the signing was successful, so I'll need to construct a dnssec-aware local name resolver.

I'd like to complete this exercise without calling for any expert assistance, and, if possible, restrict myself to online documentation during this exercise. After all, if DNSSEC really is ready for deployment then installation of DNSSEC-aware tools and services should be a mundane exercise in following the documentation. As my primary guide through the steps of DNSSEC configuration using BIND I'll be using the latest version I can locate of a RIPE tutorial document, "DNSSEC HOWTO", <http://www.ripe.net/disi/dnssec_howto/dnssec_howto.pdf> . I'm using version 1.7 of this document, from April 2005.

My platform operating system environment is FreeBSD, and I'm currently running version 6.1 of this operating system. I'll be using the BIND implementation of DNS for both the resolver tools and library, and the BIND name server.

Step 1 - Getting Bind

As of September 2006 the latest BIND version is 9.3.2-P1, and its obtainable from the Internet Software Consortium <http://www.isc.org>. The source code for this release is at <http://ftp.isc.org/isc/bind9/9.3.2-P1/bind-9.3.2-P1.tar.gz>. This code pack also appears in the FreeBSD Ports collection (http://www.freebsd.org/cgi/cvsweb.cgi/ports/dns/bind9/), so in this case I'll use the FreeBSD ports version, and follow the instructions associated with installation of this application.

Step 2 – Installing Bind9

I notice in looking through the Makefile for Bind there is a reference to OpenSSL:

```
.if defined(WITH_OPENSSL_PORT)
CONFIGURE_ARGS+=        --with-openssl=${LOCALBASE}
.else
```

```
CONFIGURE_ARGS+=        --with-openssl
.endif
```

So I'll install OpenSSL before moving on to Bind

```
# cd /usr/ports/security/openssl
# make
===>  Vulnerability check disabled, database not found
=> openssl-0.9.8c.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
=> Attempting to fetch from http://www.openssl.org/source/.
openssl-0.9.8c.tar.gz                         100% of 3236 kB  119 kBps
===>  Extracting for openssl-0.9.8c
=> MD5 Checksum OK for openssl-0.9.8c.tar.gz.
=> SHA256 Checksum OK for openssl-0.9.8c.tar.gz.
…
#make install
…
# /usr/local/bin/openssl version
OpenSSL 0.9.8c 05 Sep 2006
```

And on to installing BIND9

```
#cd /usr/lports/dns/bind9
# make install

…

*************************************************************************
*           _  _____ _____ _____ _   _ _____ ___ ___  _   _             *
*          / \|_   _|_   _| ____| \ | |_   _|_ _/ _ \| \ | |            *
*         / _ \ | |   | | |  _| |  \| | | |  | | | | |  \| |            *
*        / ___ \| |   | | | |___| |\  | | |  | | |_| | |\  |            *
*       /_/   \_\_|   |_| |_____|_| \_| |_| |___\___/|_| \_|            *
*                                                                       *
*       BIND 9 requires a good source of randomness to operate.         *
*       It also requires configuration of rndc, including a             *
*       "secret" key.  If you are using FreeBSD 4.x, visit              *
*       http://people.freebsd.org/~dougb/randomness.html for            *
*       information on how to set up entropy gathering. Users           *
*       of FreeBSD 5.x or later do not need to do this step. If         *
*       you are running BIND 9 in a chroot environment, make            *
*       sure that there is a /dev/random device in the chroot.          *
*                                                                       *
*       The easiest, and most secure way to configure rndc is           *
*       to run 'rndc-confgen -a' which will generate the proper         *
*       conf file, with a new random key, and appropriate file          *
*       permissions.                                                    *
*                                                                       *
**************************************************************************
===>   Compressing manual pages for bind9-base-9.3.2.1
===>   Registering installation for bind9-base-9.3.2.1
===> SECURITY REPORT:
      This port has installed the following files, which may act as network
      servers and may therefore pose a remote security risk to the system.
/usr/sbin/named-checkconf
/usr/sbin/rndc
/usr/sbin/lwresd
/usr/bin/nsupdate
/usr/bin/dig
/usr/sbin/named
/usr/bin/host
/usr/sbin/dnssec-signzone
/usr/bin/nslookup
/usr/sbin/named-checkzone

      If there are vulnerabilities in these programs there may be a security
      risk to the system. FreeBSD makes no guarantee about the security of
      ports included in the Ports Collection. Please type 'make deinstall'
      to deinstall the port if this is a concern.

      For more information, and contact details about the security
      status of this software, see the following webpage:
http://www.isc.org/index.pl?/sw/bind/bind9.3.php
```

Looks like I also need to configure a random sources as well

```
# cd /etc/namedb
# rndc-confgen -a
```

```
Write key file “/etc/namedb/rndc.key”
```

Now to configure BIND. The first task is to configure named.conf.. It looks like the Bind installation has provided a handy shell script to generate the localhost zone files for IPv4 and IPv4:

```
# /bin/sh make-localhost
# ls ./master
localhost-v6.rev        localhost.rev
```

I'll use a relatively minimal configuration in the first instance and just configure up a local resolver, with DNSSEC enabled. Here's `/etc/namedb/named.conf`. The "`dnssec-enable`" option I found in "`man named.conf`". At this stage I'll omit the other two DNSSEC options, "`dnssec-lookaside`" and "`dnssec-must-be-secure`". The logging command I found in the DNSSEC tutorial notes.

```
# cat named.conf

options {
  directory "/etc/namedb";
  pid-file  "/var/run/named/pid";
  listen-on { any; } ;
  listen-on-v6   { any; };
      dnssec-enable  yes ;
  };

zone "."
  { type hint; file "named.root"; };
zone "0.0.127.IN-ADDR.ARPA"
  { type master; file "master/localhost.rev"; };
zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA"
  { type master; file "master/localhost-v6.rev"; };


logging {
  channel dnssec_log {
    file "/var/log/named-dnssec.log" size 20m;
    print-time yes;
    print-category yes;
    print-severity yes;
    severity debug 10;
    };
  category dnssec { dnssec_log ; };
  };
```

Now I don't have any trust anchors configured at this stage, so not much should be happening in my local resolver relating to DNSSEC.. So lets start up thelocal resolver:

```
# /usr/sbin/named -c /etc/namedb/named.conf -d 3
# ps axuw | grep named | grep -v grep
root  29570  0.0  0.3  4076  3268  ??  Ss    3:40PM   0:00.08 /usr/sbin/named -c
/etc/namedb/named.conf -d 3
```

And the log file also shows that named has started up:

```
named[29570]: starting BIND 9.3.2-P1 -c /etc/namedb/named.conf -d 3
named[29570]: command channel listening on 127.0.0.1#953
named[29570]: command channel listening on ::1#953
named[29570]: running
```

Step 3 – A DNSSEC-aware Local Resolver

Now I'll test a retrieval of the DNSKEY RR from a signed domain. This should fail validation as I have not configured any trusted keys at this stage.

```
# dig +dnssec DNSKEY se. @127.0.0.1

; <<>> DiG 9.3.2-P1 <<>> +dnssec DNSKEY se. @127.0.0.1
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55063
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 10, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;se.                            IN      DNSKEY

;; ANSWER SECTION:
se.                    3600    IN      DNSKEY  256 3 5 AQOtFn92ppHBGVcCKDMJQ+KZE+NOowsHbjPB
                                                       4mKruGCzVHkRdlk0gu7W jgjpw/5mV0XZeI5
                                                       zKMt1IEXwQ6MvrLycshUN3SDlm/hL6e9LJBC
                                                       dyd2BVSqLAxs4eG1YPk5SzBGVeKSNMxYjekn
                                                       o/BT7cj38nCDb1PLusP/wRLdsOYIG7Q==
se.                    3600    IN      DNSKEY  256 3 5 AQPH4Rkyd28gj8v445LECVuJyRMEc+S2evIq
                                                       XHOtivUBDe0OoT44rmt5HP3ZGKXCNwa1NjOZ
                                                       pJwak9AXiK4Vmht+8cz0dEg8FdDR36bn7Qn5
                                                       V1Nf Jzg9GRKRFMbt/pvfkFdW/aELVbaRz5O
                                                       R6okHFpdTstS3aQhkUSnEG7LyjQSXlQ==
se.                    3600    IN      DNSKEY  256 3 5 AQPhX2HyqyQ/hZSo+Ra2l2q10th6/3f34L42
                                                       +vXJImGduyp0UANYAKdT6x/ct0M9HgX2wh38
                                                       ZTB1T1vtxGmVMnHNwE6JSzZM9wOfAwnNh7aS
                                                       jzOWWsjlJkm6c+Q+5e3Suy05H2NnYxgZXnWU
                                                       ChGpgDA5tTYD9p55VH6qF7okHAddew==
se.                    3600    IN      DNSKEY  256 3 5 AQPrfmP4Rnc6pFUuHwdiAsCsAtlIhhgG3+3k
                                                       +52jE5507LAJw7ve9vv7he2yNrQIHy1clR/m
                                                       IGAwpOijNxAy0H1eNHrj7xzXBAqZda7bWrFZ
                                                       ON39NGsNUVfFEpOqoD6+WgMh4aesIjvZ2tY2
                                                       MGBmLZuHQDtp6SKcV6Ty6ZIZrrx11w==
se.                    3600    IN      DNSKEY  257 3 5 AwEAAaxPMcR2x0HbQV4WeZB6oEDX+r0QM65K
                                                       bhTjrW1ZaARmPhEZZe3Y9ifgEuq7vZ/zGZUd
                                                       EGNWy+JZzus0lUptwgjGwhUS1558Hb4JKUbb
                                                       OTcM 8pwXlj0EiX3oDFVmjHO444gLkBOUKUf
                                                       /mC7HvfwYH/Be22GnClrinKJp1Og4ywzO9Wg
                                                       lMk7jbfW33gUKvirTHr25GL7STQUzBb5Usxt
                                                       8lgnyTUHs 1t3JwCY5hKZ6CqFxmAVZP20igT
                                                       ixin/1LcrgX/KMEGd/buvF4qJCyduieHukuY
                                                       3H4XMAcR+xia2nIUPvm/oyWR8BW/hWdzOvnS
                                                       CThlHf3xiYleDb t/o1OTQ09A0=
se.                    3600    IN      RRSIG   DNSKEY 5 1 3600 20060915054256 20060909010558
                                                       32327 se.
                                                       mc5o/4wR1tfugzjMT8Kyh08P2HcZSKdg1hzA
                                                       VRqFGxiBt6+xtm6JkiYuiKLxFkh9SG/Efx+v
                                                       ajuQa3aGQFPo5GSuEIL/+4OUHLo0Efz/Fu+Z
                                                       rLEl IuNYcKjRshazRmgXPrZP8mS2Ykp7lVs
                                                       fRnGUxlF1utN6bivvdiqm81KP174=
se.                    3600    IN      RRSIG   DNSKEY 5 1 3600 20061014000000 20060831093257
                                                       17686 se.
                                                       dS9hM5y4z/tWc5MPAAtireidpfLdu+oKzc60
                                                       idV7LEt5GYN1aOD9RcGj OFZMZOgu4zyR7ri
                                                       W/FovBB4anGtzIYXbxcpsqSRYSOF0Cikmlp1
                                                       vYa9GQbffUQBau8GjckdSU/I8LsGiAiPIPm9
                                                       XISOa8oYZo6eVCTPtCiXEJaRYuwoEAUJ8xQ3
                                                       aKBxPyPqoIfX96o5YOU0WOtwdASO8NIXG897
                                                       Yp1FnDIazPhn5guWSraoovn3MG4jvM3ruBpn
                                                       Lzv3ZZVew4rWZ7RXmG76mwg461vgjhI+3cQf
                                                       fzAiffIqr+DSx3Le2dnJbsSESGKRMrTeOgf8
                                                       78niBGw0rOajVbFUmIQ==

;; AUTHORITY SECTION:
se.                    172800  IN      NS      a.ns.se.
se.                    172800  IN      NS      b.ns.se.
se.                    172800  IN      NS      c.ns.se.
se.                    172800  IN      NS      d.ns.se.
se.                    172800  IN      NS      e.ns.se.
se.                    172800  IN      NS      f.ns.se.
se.                    172800  IN      NS      g.ns.se.
se.                    172800  IN      NS      h.ns.se.
se.                    172800  IN      NS      i.ns.se.
se.                    172800  IN      RRSIG   NS 5 1 172800 20060916234318 20060910050559 32327 se.
                                                       0pfexJaa98S1Fd0wEmldC687K1OOCW0oKMGOhOhMTaR
                                                       KEOSmVI1WvDsRAEtT4xjs71orLzjiX0R8e33Qux9nDs
                                                       6E811OF5W4Hfce2dSURCVdvMUiqIeB21Roq/iyOjJik
                                                       vFhlMF2feX6+OjzFnDokA3SDNemWUKZjcX7LUwR PLg=

;; Query time: 471 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Sep 10 15:41:29 2006
;; MSG SIZE  rcvd: 1652
```

The critical line of output here was:

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 10, ADDITIONAL: 1
```

The “ad” flag (“ad stands for “Authenticated Data” – which appears to be a rather important flag for DNSSEC!) is missing from the response flags. I expected this, as the configuration file omitted to load any trust keys.

This manual configuration of trust keys is perhaps the most frustrating and broken part of DNSSEC, and I suspect is the characteristic that will be the major factor in the demise of DNSSEC, if it turns out that DNSSEC fails to gain self-sustaining levels of deployment. The DNS root is not signed, nor are many, if not most, of the top level domains immediately under the root. And yet, somehow, I need to tell my resolver what zone keys should be trusted as roots of a trust model. Frankly, I don’t have a clue as to which key values I should trust and which I should reject. So to make the resolver work I’ll take an amazingly insecure short cut and place the DNSKEY for the .se domain I just retrieved into my named configuration file.

```
# tail named.conf
trusted-keys {
"se." 257 3 5 "AwEAAaxPMcR2xOHbQV4WeZB6oEDX+rOQM65KbhTjrW1ZaARmPhEZZe3Y
               9ifgEuq7vZ/zGZUdEGNWy+JZzus0lUptwgjGwhUS1558Hb4JKUbbOTcM
               8pwXlj0EiX3oDFVmjHO444gLkBOUKUf/mC7HvfwYH/Be22GnClrinKJp
               1Og4ywzO9WglMk7jbfW33gUKvirTHr25GL7STQUzBb5Usxt8lgnyTUHs
               1t3JwCY5hKZ6CqFxmAVZP20igTixin/1LcrgX/KMEGd/buvF4qJCydui
               eHukuY3H4XMAcR+xia2nIUPvm/oyWR8BW/hWdzOvnSCThlHf3xiYleDb
               t/o1OTQ09A0=";
};
```

Once more I’ll start up the resolver daemon:

```
# /usr/sbin/named –c /etc/namedb/named.conf
# tail /var/log/messaged
named[29605]: starting BIND 9.3.2-P1 -c /etc/namedb/named.conf -d 3
named[29605]: command channel listening on 127.0.0.1#953
named[29605]: command channel listening on ::1#953
named[29605]: running
```

Now lets perform a test on .se.

```
# dig +dnssec +multiline DNSKEY se. @127.0.0.1

; <<>> DiG 9.3.2-P1 <<>> +dnssec +multiline DNSKEY se. @127.0.0.1
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45272
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;se.                    IN DNSKEY

;; ANSWER SECTION:
se.                     3600 IN DNSKEY 256 3 5 (
                                AQOtFn92ppHBGVcCKDMJQ+KZE+N0owsHbjPB4mKruGCz
                                VHkRdlk0gu7Wjgjpw/5mV0XZeI5zKMt1IEXwQ6MvrLyc
                                shUN3SDlm/hL6e9lJBCdyd2BVSqLAxs4eG1YPk5SzBGV
                                eKSNMxYjekno/BT7cj38nCDb1PLusP/wRLdsOYIG7Q==
                                ) ; key id = 17474
se.                     3600 IN DNSKEY 256 3 5 (
                                AQPH4Rkyd28gj8v445LECVuJyRMEc+S2evIqXHOtivUB
                                De0OoT44rmt5HP3ZGKXCNwa1NjOZpJwak9AXiK4Vmht+
                                8cz0dEg8FdDR36bn7Qn5V1NfJzg9GRKRFMbt/pvfkFdW
                                /aELVbaRz5OR6okHFpdTstS3aQhkUSnEG7LyjQSXlQ==
                                ) ; key id = 54245
se.                     3600 IN DNSKEY 256 3 5 (
                                AQPhX2HyqyQ/hZSo+Ra2l2q1Oth6/3f34L42+vXJImGd
                                uyp0UANYAKdT6x/ct0M9HgX2wh38ZTB1T1vtxGmVMnHN
                                wE6JSzZM9wOfAwnNh7aSjzOWWsjlJkm6c+Q+5e3Suy05
                                H2NnYxgZXnWUChGpgDA5tTYD9p55VH6qF7okHAddew==
                                ) ; key id = 32327
se.                     3600 IN DNSKEY 256 3 5 (
                                AQPrfmP4Rnc6pFUuHwdiAsCsAtlIhhgG3+3k+52jE55O
                                7LAJw7ve9vv7he2yNrQIHy1clR/mIGAwpOijNxAy0H1e
                                NHrj7xzXBAqZda7bWrFZON39NGsNUVfFEpOqoD6+WgMh
                                4aesIjvZ2tY2MGBmLZuHQDtp6SKcV6Ty6ZIZrrx11w==
                                ) ; key id = 20825
se.                       3600 IN DNSKEY 257 3 5 (
                                AwEAAaxPMcR2x0HbQV4WeZB6oEDX+r0QM65KbhTjrW1Z
                                aARmPhEZZe3Y9ifgEuq7vZ/zGZUdEGNWy+JZzus0lUpt
                                wgjGwhUS1558Hb4JKUbbOTcM8pwXlj0EiX3oDFVmjHO4
                                44gLkBOUKUf/mC7HvfwYH/Be22GnClrinKJp1Og4ywzO
                                9WglMk7jbfW33gUKvirTHr25GL7STQUzBb5Usxt8lgny
                                TUHs1t3JwCY5hKZ6CqFxmAVZP20igTixin/1LcrgX/KM
                                EGd/buvF4qJCyduieHukuY3H4XMAcR+xia2nIUPvm/oy
                                WR8BW/hWdzOvnSCThlHf3xiYleDbt/o1OTQ09A0=
                                ) ; key id = 17686
se.                       3600 IN RRSIG DNSKEY 5 1 3600 20060915054256 (
                                20060909010558 32327 se.
                                mc5o/4wR1tfugzjMT8Kyh08P2HcZSKdg1hzAVRqFGxiB
                                t6+xtm6JkiYuiKLxFkh9SG/Efx+vajuQa3aGQFPo5GSu
                                EIL/+4OUHLo0Efz/Fu+ZrLElIuNYcKjRshazRmgXPrZP
                                8mS2Ykp7lVsfRnGUxlF1utN6bivvdiqm81KP174= )
se.                       3600 IN RRSIG DNSKEY 5 1 3600 20061014000000 (
                                20060831093257 17686 se.
                                dS9hM5y4z/tWc5MPAAtireidpfLdu+oKzc60idV7LEt5
                                GYN1aOD9RcGjOFZMZOgu4zyR7riW/FovBB4anGtzIYXb
                                xcpsqSRYSOF0CikmlplvYa9GQbffUQBau8GjckdSU/I8
                                LsGiAiPIPm9XISOa8oYZo6eVCTPtCiXEJaRYuwoEAUJ8
                                xQ3aKBxPyPqoIfX96o5YOU0WOtwdASO8NIXG897Yp1Fn
                                DIazPhn5guWSraoovn3MG4jvM3ruBpnLzv3ZZVew4rWZ
                                7RXmG76mwg461vgjhI+3cQffzAiffIqr+DSx3Le2dnJb
                                sSESGKRMrTeOgf878niBGw0rOajVbFUmIQ== )

;; Query time: 423 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Sep 10 15:45:30 2006
;; MSG SIZE  rcvd: 1343
```

The ad flag is set in the answer section, indicating that the answer was authenticated. Success! What did the DNSSEC debug log make of this? It seems that the local resolver had managed to decide to trust this key.

```
# tail /var/log/named-dnssec.log
 validating @0x824c000: se DNSKEY: starting
 validating @0x824c000: se DNSKEY: attempting positive response validation
 validating @0x824c000: se DNSKEY: verify rdataset: success
 validating @0x824c000: se DNSKEY: signed by trusted key; marking as secure
 validator @0x824c000: dns_validator_destroy
```

Now who else has published keys that I may want to configure? This may be a fine question, but, once more, I have no idea what the answer should be. It seems that this technology was intended to be a top-down comprehensively signed structure, where all I would need use to seed DNSSEC was the current key for the DNS root, and all other DNS keys could be validated by performing a backward walk towards the root. And for as long as the root remains unsigned, and for as long as the next level down, the top level domains remain unsigned, then anyone attempting to perform DNSSEC checks on resolver outcomes is pretty much indulging in a pointless exercise.

So right now I have a DNSSEC-aware resolver, and because I have decided to dynamically load a DNS zone key arbitrarily, I can perform key resolution on a limited set of domains under .se. Precisely which set of domains under .se can be validated using DNSSEC I can't tell in advance. Some work in terms of validation, some do not. I cannot tell if a response for a query relating to a sub-domain of .se should have additional authentication data or not. As far as I can tell this is not a terribly useful outcome, so I'll spend a little time looking around for some more trust anchors. The RIPE NCC publish a set of trust anchors at <https://www.ripe.net/projects/disi//keys/>. I'll load these as trust anchors into my `named.conf` file and test these:

```
# dig +dnssec www.ripe.net A @127.0.0.1

; <<>> DiG 9.3.2-P1 <<>> +dnssec www.ripe.net A @127.0.0.1
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51789
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
```

```
;; QUESTION SECTION:
;www.ripe.net.                  IN      A

;; ANSWER SECTION:
www.ripe.net.           600     IN      CNAME   kite-www.ripe.net.
www.ripe.net.           600     IN      RRSIG   CNAME 5 3 600 20061010051144 20060910051144 15917
ripe.net. BcSb934sDbz8GiouhjT8zOAum5v1hBq7+82JhXCucdORFsAqHtkIcRSn
Oxhp89Mu/EbugkRBFbAzoV6ljBhRODL1/jLD2pQqyZ8jWWre4ElcAo7s
wE/xSoIrrutwyA5RB09OdtGINX0JbarkaE9LBVQb1v0MDKdGjpNzoNPW WVxsZ7ioddqCJax5+r7W+q1HlmhSKU8f
kite-www.ripe.net.      3600    IN      A       193.0.0.214
kite-www.ripe.net.      3600    IN      RRSIG   A 5 3 172800 20061010051144 20060910051144 15917
ripe.net. OBFOlty7FDxApqEMBGC0MbqgFP+ePVKeL/ZLlBmV9j6j1B97U++/6NlQ
pqw8PJYx826y/7tCsT3LOm3dnzJdVkcE9No/VPtB/kyRcQLRlYnSL4F8
gT5sw5H7NBD4+w8orwfm1GG9Lqmff2q0rZU7tiRP1i7U9/ASLTj2i/9H D+ASg/lye66pcLdOoPQVCNnfCL1m9GfH

;; Query time: 1657 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Sep 11 10:56:07 2006
;; MSG SIZE  rcvd: 460
```

Again the “ad” flag bit appears to be set, so this response appears to be valid. How about a non-existent name within the ripe.net zone?

```
# dig +dnssec A nonexistent.ripe.net @127.0.0.1

; <<>> DiG 9.3.2-P1 <<>> +dnssec A nonexistent.ripe.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 42805
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;nonexistent.ripe.net.          IN      A

;; AUTHORITY SECTION:
ripe.net.               3600    IN      SOA     ns-pri.ripe.net. ops.ripe.net. 2006091101 43200 7200
1209600 7200
ripe.net.               3600    IN      RRSIG   SOA 5 2 172800 20061011051206 20060911051206 15917
ripe.net. jKMONWMlg4YnzZyotmcRJ9VSbHdNTkEYdbrV/RYnq9urgu8DP0ca6TSC
rZQpwEaMSth4PkHfcfo9hx3uKCP1Hcl+Dx8U91KwWZQ5s+8o3FAW76I9
FhZLQmg9GVDYU2O2VAcGWvmuYFe0uhI6Br2d7h3zSGivkax81Vd4OIgD sGA2sb2NAcT+TF3SmOots01JJPhs1VfW
ripe.net.               3600    IN      NSEC    adsl.ripe.net. A NS SOA MX RRSIG NSEC DNSKEY
ripe.net.               3600    IN      RRSIG   NSEC 5 2 7200 20061011051206 20060911051206 15917
ripe.net. koQhABkByga7YCGwu6pGuqik9y1j4aeeRU9gbkpA7YJULjpYjAnFv0dG
+4xBydIT7jT/9N0Yf9OpKRdp3Cgv8ZKTMflyD/rkyxWQXYujzhPnLS1H
iOwv+WjhRMv5oJ7HmZSlCrcJ34zq+SWmHKuCOH60hnray7eTTpQqt1dd +ptjZsXCtw2bltYO826a2ukRrD8HRIKj
niletest.ripe.net.      3600    IN      NSEC    np-console.ripe.net. A RRSIG NSEC
niletest.ripe.net.      3600    IN      RRSIG   NSEC 5 3 7200 20061011051206 20060911051206 15917
ripe.net. wr1LMfpPhtaIxe1Kcx/SSmB/Wq9cez+DeyO0PX05ido2nMD8pdpSQzGs
168YOol5JaC0fP6dgjtTFq3AqabF+egPJwmQbJS2hRGYPXWq74UQS81L
SG2hs6Bjorj/MNNL/5eMoPrDeG8lALTIcavfGfIGoet0ToKwaOSU+bEp 0qK3/LMUuQ9iesUzUx0x8KWoR+fG/Mt9

;; Query time: 1112 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Sep 12 08:11:14 2006
;; MSG SIZE  rcvd: 752
```

The “ad” bit is set, which tells me that the answer came from the original zone file, and the relevant NSEC record tells me that there are no names between `niletest.ripe.net` and `np-console.ripe.net`. How about querying `nlnetlabs.nl`?

```
# dig +dnssec DNSKEY nlnetlabs.nl @127.0.0.1

; <<>> DiG 9.3.2-P1 <<>> +dnssec DNSKEY nlnetlabs.nl @127.0.0.1
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24068
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;nlnetlabs.nl.                  IN      DNSKEY

;; ANSWER SECTION:
```

```
nlnetlabs.nl.           86386   IN      DNSKEY  257 3 5
AQPzzTWMz8qSWIQlfRnPckx2BiVmkVN6LPupO3mbz7FhLSnm26n6iG9N
Lby97Ji453aWZY3M5/xJBSOS2vWtco2t8C0+xeO1bc/d6ZTy32DHchpW
6rDH1vp86Ll+ha0tmwyy9QP7y2bVw5zSbFCrefk8qCUBgfHm9bHzMG1U BYtEIQ==
nlnetlabs.nl.           86386   IN      RRSIG   DNSKEY 5 2 86400 20060902172237 20060803172237 43791
nlnetlabs.nl. PcxHlUzBwPgxq7uVzkjgN5Kl3Z4yq3Ie57O8TQu3A2w/J+2/eLQdbBTC
R9yld6We2YtRTwNfPumdUvFQWTMuRj6tg8TVggSYVv/uHs2ySC24NdpT
9QXsMwJnKmj59DmhroLvh8svHnZ7W1I8y+tvs7pWWE2tw0KF+pPKYVk1 61g=

;; AUTHORITY SECTION:
nlnetlabs.nl.           86386   IN      NS      open.nlnetlabs.nl.
nlnetlabs.nl.           86386   IN      NS      omval.tednet.nl.
nlnetlabs.nl.           86386   IN      NS      ns7.domain-registry.nl.
nlnetlabs.nl.           86386   IN      RRSIG   NS 5 2 86400 20060902172237 20060803172237 43791
nlnetlabs.nl. fcoLufPkFtOhIzDp6cvT4ZXeFnHAiD0sWz1OWr5cnSDW9/TKmEiPeRyY
sb0HocNNiTg1Gl+wziVWlMEK87CJYi1l+E3lXOLZr0fz/l4lzOiZIr7N
DEwwYMGLyFopWpKh3ThJoiotVktEkJHOzF/Kie+OZHFLoIfcf8dVM0Tu ntg=

;; Query time: 17 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Sep 11 11:05:41 2006
;; MSG SIZE  rcvd: 611
```

Actually this one looks like it was never going to work – the dates on the RRSIG Resource Records indicate key expiry on the 2nd September, and I'm conducting this test on the 11th of September. The key has expired for this zone.

So it looks like the DNSSEC resolver works, after a fashion, although I have to say that the treatment of trust anchors is relatively useless for all but the most trivial of exercises.

Can I do anything better in terms of discovery of trust keys in the absence of a comprehensively signed DNS structure starting at the root? It appears that RFC4431 contains a potential answer to this problem of fragmented trust anchors. This document describes a DNSSEC Lookaside Validation (DLV) DNS Resource Record. In this approach the trust anchors are retrieved from a lookaside location, and by configuring the credentials of the DLV publisher into the resolver, then the resolver will be aware to recognise as trust acnhors all those zones that have registered with the DLV publisher. The configuration details for the ISC-operated DLV service are published at <http://www.isc.org/index.pl?/ops/dlv/.>. Using the configuration steps at that URL my `named.conf` now has an additional trusted key, this one for dlv.isc.org.

So into the trusted-keys section of `named.conf` I've added

```
dlv.isc.org. 257 3 5
"AQPap3+2+itqZpuujLA/j/eIEyls9HGo9W8rm1uVpW0zZX4viyFQyGL91YkGUA2uTQ1ZHWbJ36KYlJpt8ZZ+
tuIismJw9/AUnNzlPgwCfq5C2MOGVh33nF6Ok67ppiapMYsOaDFbAQf5Vcc3L+BwfJvkXsZK73nD3gBEcdcmuJejeQ==";
```

and into the options section I've added

```
dnssec-lookaside . trust-anchor dlv.isc.org;
```

Now - how do I know this lookaside is working? I don't! I don't know who has lodged their zone keys with the operators of this service, nor can I see from any of the online information where such information can be obtained. So how am I to know how to test this lookaside service? Which domains have I implicitly added to my trusted set? It seems to me that there is some critical information that is lacking here, and in its current state the lookaside exercise is one that looks rather like one of futility.

So in the absence of any decent details, lets try a guess - if this lookaside service, operated by ISC, is working, then I should expect that a query for the A record of www.isc.org should validate via this lookaside mechanism - right?

Wrong.

```
# dig +dnssec A www.isc.org
```

```
; <<>> DiG 9.3.2-P1 <<>> +dnssec A www.isc.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15499
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.isc.org.                   IN      A

;; ANSWER SECTION:
www.isc.org.            600     IN      A       204.152.184.88
www.isc.org.            600     IN      RRSIG   A 5 3 600 20061010193344 20060910193344 57956
isc.org. T/IvQDOmTrmf7fKAJ8gkRH4t/J2zgCx4YPkLzjPAnexJbjsxrTkDq/uo
WgLO5OsXHmHxVTPwZA96+Q+oFkoq2/MXBiYbnM3U1xTfFTFWT3fcuxmL
Hc6kCa9UbD/7hPwt/VrvB38Y9pHLeDj7Ehr8+EfaROcdJzuvpfSEN/SS 084=

;; Query time: 1593 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Sep 12 08:01:02 2006
;; MSG SIZE  rcvd: 223

#
```

The “ad” flag is missing in the response. What went wrong?

The log file relating to this query indicates that dlv.isc.org appears to validate, but that www.isc.org is not included in the DLV service (or at least that’s my interpretation of the following voluminous debug output):

```
 validating @0x824b000: . NS: starting
 validating @0x824b000: . NS: looking for DLV
 validating @0x824b000: . NS: plain DNSSEC returns unsecure (.): looking for DLV
 validating @0x824b000: . NS: looking for DLV dlv.isc.org
 validating @0x824b000: . NS: finddlvsep: creating fetch for dlv.isc.org DLV
 validating @0x824b000: . NS: DLV lookup: wait
 validating @0x824b800: dlv.isc.org DLV: starting
 validating @0x824b800: dlv.isc.org DLV: attempting negative response validation
 validating @0x824b800: dlv.isc.org DLV: nsecvalidate: creating validator for dlv.isc.org SOA
   validating @0x824f000: dlv.isc.org SOA: starting
   validating @0x824f000: dlv.isc.org SOA: attempting positive response validation
   validating @0x824f000: dlv.isc.org SOA: get_key: creating fetch for dlv.isc.org DNSKEY
 validating @0x824f800: www.isc.org A: starting
 validating @0x824f800: www.isc.org A: looking for DLV
 validating @0x824f800: www.isc.org A: plain DNSSEC returns unsecure (.): looking for DLV
 validating @0x824f800: www.isc.org A: looking for DLV www.isc.org.dlv.isc.org
 validating @0x824f800: www.isc.org A: finddlvsep: creating fetch for www.isc.org.dlv.isc.org DLV
 validating @0x824f800: www.isc.org A: DLV lookup: wait
 validating @0x825a000: dlv.isc.org DNSKEY: starting
 validating @0x825a000: dlv.isc.org DNSKEY: attempting positive response validation
 validating @0x825a000: dlv.isc.org DNSKEY: verify rdataset: success
 validating @0x825a000: dlv.isc.org DNSKEY: signed by trusted key; marking as secure
 validator @0x825a000: dns_validator_destroy
   validating @0x824f000: dlv.isc.org SOA: in fetch_callback_validator
   validating @0x824f000: dlv.isc.org SOA: keyset with trust 7
   validating @0x824f000: dlv.isc.org SOA: resuming validate
   validating @0x824f000: dlv.isc.org SOA: verify rdataset: success
   validating @0x824f000: dlv.isc.org SOA: marking as secure
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: starting
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: attempting negative response validation
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: nsecvalidate: creating validator for
dlv.isc.org SOA
   validating @0x8258800: dlv.isc.org SOA: starting
   validating @0x8258800: dlv.isc.org SOA: attempting positive response validation
   validating @0x8258800: dlv.isc.org SOA: keyset with trust 7
   validating @0x8258800: dlv.isc.org SOA: verify rdataset: success
   validating @0x8258800: dlv.isc.org SOA: marking as secure
   validator @0x8258800: dns_validator_destroy
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: in authvalidated
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: resuming nsecvalidate
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: nsecvalidate: creating validator for
isc.org.dlv.isc.org NSEC
   validating @0x8258800: isc.org.dlv.isc.org NSEC: starting
   validating @0x8258800: isc.org.dlv.isc.org NSEC: attempting positive response validation
   validating @0x8258800: isc.org.dlv.isc.org NSEC: keyset with trust 7
   validating @0x8258800: isc.org.dlv.isc.org NSEC: verify rdataset: success
   validating @0x8258800: isc.org.dlv.isc.org NSEC: marking as secure
   validator @0x824f000: dns_validator_destroy
 validating @0x824b800: dlv.isc.org DLV: in authvalidated
 validating @0x824b800: dlv.isc.org DLV: resuming nsecvalidate
 validating @0x824b800: dlv.isc.org DLV: nsecvalidate: creating validator for dlv.isc.org NSEC
   validating @0x824f000: dlv.isc.org NSEC: starting
   validating @0x824f000: dlv.isc.org NSEC: attempting positive response validation
   validating @0x824f000: dlv.isc.org NSEC: keyset with trust 7
   validating @0x824f000: dlv.isc.org NSEC: verify rdataset: success
   validating @0x824f000: dlv.isc.org NSEC: marking as secure
   validator @0x824f000: dns_validator_destroy
 validating @0x824b800: dlv.isc.org DLV: in authvalidated
 validating @0x824b800: dlv.isc.org DLV: looking for relevant nsec
 validating @0x824b800: dlv.isc.org DLV: nsec proves name exists (owner) data=0
 validating @0x824b800: dlv.isc.org DLV: resuming nsecvalidate
 validating @0x824b800: dlv.isc.org DLV: nonexistence proof found
 validator @0x824b800: dns_validator_destroy
   validator @0x8258800: dns_validator_destroy
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: in authvalidated
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: looking for relevant nsec
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: nsec range ok
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: resuming nsecvalidate
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: in checkwildcard: *.isc.org.dlv.isc.org
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: looking for relevant nsec
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: nsec range ok
 validating @0x8258000: www.isc.org.dlv.isc.org DLV: nonexistence proof found
 validator @0x8258000: dns_validator_destroy
 validating @0x824b000: . NS: in dlvfetched: ncache nxrrset
 validating @0x824b000: . NS: DLV not found
 validating @0x824b000: . NS: marking as answer
 validator @0x824b000: dns_validator_destroy
 validating @0x824f800: www.isc.org A: in dlvfetched: ncache nxdomain
 validating @0x824f800: www.isc.org A: looking for DLV isc.org.dlv.isc.org
 validating @0x824f800: www.isc.org A: finddlvsep: creating fetch for isc.org.dlv.isc.org DLV
 validating @0x824f800: www.isc.org A: DLV lookup: wait
 validating @0x824f000: isc.org.dlv.isc.org DLV: starting
 validating @0x824f000: isc.org.dlv.isc.org DLV: attempting negative response validation
 validating @0x824f000: isc.org.dlv.isc.org DLV: nsecvalidate: creating validator for dlv.isc.org
SOA
   validating @0x81cd000: dlv.isc.org SOA: starting
   validating @0x81cd000: dlv.isc.org SOA: attempting positive response validation
   validating @0x81cd000: dlv.isc.org SOA: keyset with trust 7
   validating @0x81cd000: dlv.isc.org SOA: verify rdataset: success
   validating @0x81cd000: dlv.isc.org SOA: marking as secure
   validator @0x81cd000: dns_validator_destroy
 validating @0x824f000: isc.org.dlv.isc.org DLV: in authvalidated
 validating @0x824f000: isc.org.dlv.isc.org DLV: resuming nsecvalidate
 validating @0x824f000: isc.org.dlv.isc.org DLV: nsecvalidate: creating validator for
isc.org.dlv.isc.org NSEC
   validating @0x81cd000: isc.org.dlv.isc.org NSEC: starting
   validating @0x81cd000: isc.org.dlv.isc.org NSEC: attempting positive response validation
   validating @0x81cd000: isc.org.dlv.isc.org NSEC: keyset with trust 7
   validating @0x81cd000: isc.org.dlv.isc.org NSEC: verify rdataset: success
   validating @0x81cd000: isc.org.dlv.isc.org NSEC: marking as secure
   validator @0x81cd000: dns_validator_destroy
 validating @0x824f000: isc.org.dlv.isc.org DLV: in authvalidated
 validating @0x824f000: isc.org.dlv.isc.org DLV: looking for relevant nsec
 validating @0x824f000: isc.org.dlv.isc.org DLV: nsec proves name exists (owner) data=0
 validating @0x824f000: isc.org.dlv.isc.org DLV: resuming nsecvalidate
 validating @0x824f000: isc.org.dlv.isc.org DLV: nonexistence proof found
 validator @0x824f000: dns_validator_destroy
 validating @0x824f800: www.isc.org A: in dlvfetched: ncache nxrrset
 validating @0x824f800: www.isc.org A: looking for DLV org.dlv.isc.org
 validating @0x824f800: www.isc.org A: DNS_R_COVERINGNSEC
 validating @0x824f800: www.isc.org A: covering nsec: not in range
 validating @0x824f800: www.isc.org A: finddlvsep: creating fetch for org.dlv.isc.org DLV
 validating @0x824f800: www.isc.org A: DLV lookup: wait
 validating @0x824f000: org.dlv.isc.org DLV: starting
 validating @0x824f000: org.dlv.isc.org DLV: attempting negative response validation
 validating @0x824f000: org.dlv.isc.org DLV: nsecvalidate: creating validator for dlv.isc.org SOA
   validating @0x81cd000: dlv.isc.org SOA: starting
   validating @0x81cd000: dlv.isc.org SOA: attempting positive response validation
   validating @0x81cd000: dlv.isc.org SOA: keyset with trust 7
   validating @0x81cd000: dlv.isc.org SOA: verify rdataset: success
   validating @0x81cd000: dlv.isc.org SOA: marking as secure
   validator @0x81cd000: dns_validator_destroy
 validating @0x824f000: org.dlv.isc.org DLV: in authvalidated
 validating @0x824f000: org.dlv.isc.org DLV: resuming nsecvalidate
 validating @0x824f000: org.dlv.isc.org DLV: nsecvalidate: creating validator for ns-ext.dlv.isc.org
NSEC
   validating @0x81cd000: ns-ext.dlv.isc.org NSEC: starting
   validating @0x81cd000: ns-ext.dlv.isc.org NSEC: attempting positive response validation
   validating @0x81cd000: ns-ext.dlv.isc.org NSEC: keyset with trust 7
   validating @0x81cd000: ns-ext.dlv.isc.org NSEC: verify rdataset: success
   validating @0x81cd000: ns-ext.dlv.isc.org NSEC: marking as secure
   validator @0x81cd000: dns_validator_destroy
```

```
validating @0x824f000: org.dlv.isc.org DLV: in authvalidated
validating @0x824f000: org.dlv.isc.org DLV: looking for relevant nsec
validating @0x824f000: org.dlv.isc.org DLV: nsec proves name exist (empty)
validating @0x824f000: org.dlv.isc.org DLV: resuming nsecvalidate
validating @0x824f000: org.dlv.isc.org DLV: nonexistence proof found
validator @0x824f000: dns_validator_destroy
validating @0x824f800: www.isc.org A: in dlvfetched: ncache nxrrset
validating @0x824f800: www.isc.org A: looking for DLV dlv.isc.org
validating @0x824f800: www.isc.org A: DLV not found
validating @0x824f800: www.isc.org A: marking as answer
validator @0x824f800: dns_validator_destroy
```

That's a massive amount of diagnostic output for an authentication failure!

I think about I've gone about as far as I can with the resolver configuration. If I, as an operator of a DNS resolver, am prepared to spend large amounts of time to track down trust anchor keys of DNS zones, and regularly monitor their currency, and re-fetch as necessary, then I can validate a subset of the DNS. Given the relatively small amount of time I'm prepared to spend on this task I suspect that this subset of the DNS will be exceptionally small. If I am prepared to trust a lookaside service then I can potentially validate a greater fraction of the DNS name space, but, frankly, I have no idea what is behind the DLV curtain, and as a consumer who is concerned enough about the validity of the DNS to arm my resolver with DNSSEC, then in its current incarnation DLV is not doing much for me at all.

Interestingly, if I want to use DNSSEC at the application level I can't see any readily available tools at all right now. The 'standard' application level queries to the DNS are through the gethostbyname() library call, and this interface has no clear way in which to add DNSSEC validation switches. It would appear that if I want to write an application that looks at DNSSEC validation outcomes of DNS queries then I need to write my own code right down to the DNS call primitives. Personally, I liked the gethostbyname() abstraction of the interface to the DNS, and I'd strongly prefer the availability of a similar abstraction of a DNSSEC-aware name resolution call.

Even if I had a gethostbyname_with_dnssec_validation() call the situation is still not entirely clear – what should my application do if the DNSSEC validation check fails? Should it generate a popup dialogue with the user, alerting the user to the validation problem and requesting whether to proceed or not? The impression that many folk have is that the standard user response to certificate invalidity pop-up warnings is to direct the application to proceed in any case. If these were to be the case with DNSSEC pop-ups, and users would simply say "proceed" in any case then there is a real issue about the true value of DNSSEC to the end user. And, of course the DNS is used in various internal functions, where there is no clear mechanism for user alerts and intervention. If such a daemon detects a DNSSEC invalidity condition should it refuse to continue, or log the event and continue in any case? The more generic question is what should happen once you arm an application with sufficient capability to check the authenticity of information received over a network. Should the validation provide a yes/no condition for continuing with the application, or should it be interpreted as a preference, or should it be disregarded? As I don't have the time in this exercise to construct a DNSSEC-aware application this remains an open question here. If DNSSEC were universally deployed, then the answer may be clearer. Given a situation of partial deployment of DNSSEC the absence of DNSSEC credentials in a response does not necessarily mean that there has been a faked DNS response, but on the other hand you simply cannot be sure.

Step 4 – Signing a Zone

The next part of the exercise with DNSSEC is to sign a zone. I'll set up a zone just for this purpose, so I'll call it "dnssec.potaroo.net".

This zone is relatively simple:

```
$TTL 86400

$ORIGIN         dnssec.potaroo.net.
```

```
@               IN      SOA     dns0.potaroo.net. gih.potaroo.net. (
                2006090803 ; Serial
                3h   ; Refresh
                15   ; Retry
                1w   ; Expire
                3h ) ; Minimum
;
;
;
; name servers
;
                IN      NS      dns0.potaroo.net.
                IN      NS      dns1.potaroo.net.
;
; subdomains
;
sub             IN      NS      dns0.dnssec.potaroo.net.
                IN      NS      dns1.dnssec.potaroo.net.
;
; zone A records
;
www             IN      A       203.50.0.6
bgp             IN      A       203.50.0.159
bgp2            IN      A       203.50.0.33
dns0            IN      A       203.50.0.18
dns1            IN      A       203.50.0.6
;
; wildcard
;
*               IN      A       203.50.0.18
```

The first task is to generate some keys to sign the zone. I'll choose a 1024 bit key size , and use split Key Signing Leys and Zone signing keys. The first key I'll generate is the Zone signing key. I'll use the RIPE tutorial for the command syntax for this command:

```
# dnssec-keygen -r/dev/random  -a RSASHA1 -b 1024 -n ZONE dnssec.potaroo.net
Kdnssec.potaroo.net.+005+03755
```

The command has generated two files, one with the public key information in the form of a DNSKEY record, and the other is the description of the private key. Here's the public key file:

```
# cat Kdnssec.potaroo.net.+005+03755.key
dnssec.potaroo.net. IN DNSKEY 256 3 5 AQO8xvbN4hZ8bn926wpM8c9Uqqhqcf45v73k4J/YSu+6o/QsPCKwJoDY
xMH3s5ZONJlgLUQscIZZKDYVHPW3Txt59bHrn739osnQ8ORbOGVTH/Vi
//L3BGjZrZr+PWtH2Vb3wIhrujMej2m4E2Mth/XjSDAhYZVWCNhJGOnP H6G6Ww==
```

I'll repeat this process for the key signing key, adding "-f KSK" to the command:

```
# dnssec-keygen -r/dev/random  -f KSK -a RSASHA1 -b 1024 -n ZONE dnssec.potaroo.net
Kdnssec.potaroo.net.+005+29022
```

I now have four key files:

```
# ls Kdns*
Kdnssec.potaroo.net.+005+03755.key      Kdnssec.potaroo.net.+005+29022.key
Kdnssec.potaroo.net.+005+03755.private  Kdnssec.potaroo.net.+005+29022.private
```

AS per the tutorial instructions, I'll copy the public keys into the named master zone area, and include them into the dnssec.potaroo.net zone file

```
# tail /etc/namedb/master/dnssec.potaroo.net
;
; wildcard
;
*               IN      A       203.50.0.18

$include Kdnssec.potaroo.net.+005+03755.key      ; zone signing key
$include Kdnssec.potaroo.net.+005+29022.key      ; key singing key
```

Before signing the zone, maybe I should check it for syntactic correctness:

```
# named-checkzone -D dnssec.potaroo.net dnssec.potaroo.net
```

```
zone dnssec.potaroo.net/IN: loaded serial 2006090802
dnssec.potaroo.net.                           86400 IN SOA      dns0.potaroo.net. gih.potaroo.net.
2006090802 10800 15 604800 10800
dnssec.potaroo.net.                           86400 IN NS       dns0.potaroo.net.
dnssec.potaroo.net.                           86400 IN NS       dns1.potaroo.net.
dnssec.potaroo.net.                           86400 IN DNSKEY   256 3 5
AQO8xvbN4hZ8bn926wpM8c9Uqqhqcf45v73k4J/YSu+6o/QsPCKwJoDY
xMH3s5ZONJlgLUQscIZZKDYVHPW3Txt59bHrn739osnQ8ORbOGVTH/Vi
//L3BGjZrZr+PWtH2Vb3wIhrujMej2m4E2Mth/XjSDAhYZVWCNhJG0nP H6G6Ww==  ; key id = 3755
dnssec.potaroo.net.                           86400 IN DNSKEY   257 3 5
AQPSOR9BUnuQQ8ien6WibaSsKddzZstW4TEuJrSzezQL79DFqHeOvVuh
Jr+9JMQmJuQGUjVcXDG1gBRQboiFJ6e+G6sibIKlkzXCLSX7O9YqYtyv
1AMyEbYWLTwRvKojZSZr2LyKqeKGFqWdoA8a1M6XRuChBlwxMwo5I5fs edIyYw==  ; key id = 29022
*.dnssec.potaroo.net.                         86400 IN A        203.50.0.18
bgp.dnssec.potaroo.net.                       86400 IN A        203.50.0.159
bgp2.dnssec.potaroo.net.                      86400 IN A        203.50.0.33
dns0.dnssec.potaroo.net.                      86400 IN A        203.50.0.18
dns1.dnssec.potaroo.net.                      86400 IN A        203.50.0.6
sub.dnssec.potaroo.net.                       86400 IN NS       dns0.dnssec.potaroo.net.
sub.dnssec.potaroo.net.                       86400 IN NS       dns1.dnssec.potaroo.net.
www.dnssec.potaroo.net.                       86400 IN A        203.50.0.6
OK
```

Looks good enough! Now to sign the zone. Again I'll use a command format following the RIPE tutorial.

```
# /usr/local/sbin/dnssec-signzone -r /dev/random -o dnssec.potaroo.net.
         -k Kdnssec.potaroo.net.+005+29022 dnssec.potaroo.net Kdnssec.potaroo.net.+005+03755.key
dnssec.potaroo.net.signed
```

I should see a signed zone file in the file dnssec.potaroo.net.signed:

```
# cat dnssec.potaroo.net.signed
; File written on Fri Sep  8 19:08:32 2006
; dnssec_signzone version 9.3.2
dnssec.potaroo.net.     86400   IN SOA  dns0.potaroo.net. gih.potaroo.net. (
                                        2006090803 ; serial
                                        10800      ; refresh (3 hours)
                                        15         ; retry (15 seconds)
                                        604800     ; expire (1 week)
                                        10800      ; minimum (3 hours)
                                        )
                        86400   RRSIG   SOA 5 3 86400 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        syLogFkxP1KlEkYp4Pic6qgW1Nr16powlzx+
                                        VbpdA/erzxRdARd1l77F56N7TB+v3aS82aLh
                                        BLIN+f0MzHEo/JNWVl0xjn95pRDd3gyZSoE+
                                        aWG21MokMbTBxF2pYmFA1ENNKKK+pSXuXvsS
                                        dAP+kcVqT6PfO67+m2chsqbh+uA= )
                        86400   NS      dns0.potaroo.net.
                        86400   NS      dns1.potaroo.net.
                        86400   RRSIG   NS 5 3 86400 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        p2kKLK4gzlm8nkr4lpXyz4FirWWXtiyXc5X/
                                        Ns2NYC3CNYDNIRFHzEI14RZO08R9z4aoQlfO
                                        jXidiJZ2BgxzmykVJUaA7AwGirVtr+6wDJrd
                                        if9tm7UdYN2powrP9o2lq0DKhwYk8i4Dyjdd
                                        9kwt7/x44ZECzEj7w30GfW4uvy8= )
                        10800   NSEC    *.dnssec.potaroo.net. NS SOA RRSIG NSEC DNSKEY
                        10800   RRSIG   NSEC 5 3 10800 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        h75DS6C1lGLPRbqtz9+KV4oSuidA+Bdt6geq
                                        q6NRrneNGA6Rr00FK4Td9AQS1+JpM3KriDl5
                                        LKqQM7yMarC7aE3v/23iW9YqFv3Z6PpjW7Ze
                                        oEhaLNCV3kG4tVmILsoGEp/EWtgNTnXkJdkD
                                        hW+o91s7XVnGmO7m9JkUOu8sS2E= )
                        86400   DNSKEY  256 3 5 (
                                        AQO8xvbN4hZ8bn926wpM8c9Uqqhqcf45v73k
                                        4J/YSu+6o/QsPCKwJoDYxMH3s5ZONJlgLUQs
                                        cIZZKDYVHPW3Txt59bHrn739osnQ8ORbOGVT
                                        H/Vi//L3BGjZrZr+PWtH2Vb3wIhrujMej2m4
                                        E2Mth/XjSDAhYZVWCNhJG0nPH6G6Ww==
                                        ) ; key id = 3755
                        86400   DNSKEY  257 3 5 (
                                        AQPSOR9BUnuQQ8ien6WibaSsKddzZstW4TEu
                                        JrSzezQL79DFqHeOvVuhJr+9JMQmJuQGUjVc
                                        XDG1gBRQboiFJ6e+G6sibIKlkzXCLSX7O9Yq
                                        Ytyv1AMyEbYWLTwRvKojZSZr2LyKqeKGFqWd
                                        oA8a1M6XRuChBlwxMwo5I5fsedIyYw==
                                        ) ; key id = 29022
                        86400   RRSIG   DNSKEY 5 3 86400 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        EMXe2OwX8CNOeAg1iexEMSlGUuApeIB/zW1z
                                        pHhZ+I/9YFE2bmmWaj6+jtfMMW8tvjlqdEFH
                                        8TOihsMaPhu0nMQnqTrKTNS4Y4DkHqt05N6a
                                        3yS1h/ufRfBDn2rA5EquVNGZM6TRI0iweDSn
                                        1HsWy5+FiQcCFubsVJjCyqG/RXo= )
                        86400   RRSIG   DNSKEY 5 3 86400 20061008080832 (
                                        20060908080832 29022 dnssec.potaroo.net.
                                        plmpAtyiQINPi0RcibIcry9eofJhvm6mkxZS
                                        nL5Qb4x/g+DC02kXMhFCsVvNSU9ATAwRIOhY
                                        PG85LaC7FdfWdOud5I+AVvVPRB+8aX1scS/8
                                        /kQ5AbJuxT3b6ezCEhu2FSuRKN3uskV5Af4N
                                        1nBBVmFWd7vXR53Q6KCucWjBvmg= )
*.dnssec.potaroo.net.   86400   IN A    203.50.0.18
                        86400   RRSIG   A 5 3 86400 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        UTLlJPY60iaVo8skJKbljkF+Dz0ZFJiPGSLM
                                        EmmzHVlYNeIjpQNK/o5jcIdDv7S4MZ+MJg31
                                        MLPXuStBWe8ElfwU4w+eQX38dXP1fPs2Mjz2
                                        RyG/dw2krgvVRfQDa27UJVurxDxoQTykEwW7
                                        yYzAdA6oVflEkjyTF8O/CxrGVy0= )
                        10800   NSEC    bgp.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 3 10800 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        ThBINgb7kHEq5t+wunmN/uTxlE5Z3nxI29e9
                                        eFFidmBmMo459/oXeuc8w8kh9U0X2TQ1og8L
                                        3GQwLNO75JrbsgMOSGzhNVD5b7Yj7PZNPWa7
                                        M4O8z7ok3Dru5XOYf4NV5fORUsvHhBnOBr+/
                                        6wTSdnpI/mQvGk5EmCKPwkvhzqE= )
bgp.dnssec.potaroo.net. 86400   IN A    203.50.0.159
                        86400   RRSIG   A 5 4 86400 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        BjQFfiLaiOxP4rKIzT9OvteuVR3kR2NBYZgM
                                        WMQxbSK6X4b84hE6HTRparY71bXXBvcKXIt7
                                        MpWX97m1A9KScR7b37h084ZE1I6b86eaN3f9
                                        Ad+9X1NXPw/RdrQZXby5xkyNSB0oIpM8R0Jz
                                        kKGGi+OO5tn7O3TyBWMrlCznlaA= )
                        10800   NSEC    bgp2.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        SjjK2OpKnv514pUd0cfTMkgpqggljvcf+1NP
                                        fizuFXMjOewJbdskKxE9FaRHwrDNvQpnwdyy
                                        adgv+TBRLZhtHr1pO7aFfPyXCnsABffnPhWc
                                        0s/xb1mAhAmPtf3f7Ri/CxrF5HFQF/lHHbHW
                                        UUHzU2dkM8wOHzkGP/OPv5oNDOo= )
bgp2.dnssec.potaroo.net. 86400  IN A    203.50.0.33
                        86400   RRSIG   A 5 4 86400 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        fuYkcujF/miEDcfSEPPAC/5wvYg6MmEqmlsg
                                        xFTwDykT0otCSdsy5R/20meDtWbYWqwI1Wb5
                                        7zTuBmSJprklTieq69j8Ptr4y3JGEsNeGA14
                                        1fDMpqdT29kgvWJHKiZyEJ7Hj2zV9WuOrpu6
                                        6hzW7pKz/xm9+Xv2ssx+u5nfrXU= )
                        10800   NSEC    dns0.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        S+X4oHey+hizmSF8d73877qYGK782uJzLgOA
                                        dHnOD9RC0SolSriKfSD+l/q+47ckBhsaMx5B
                                        9jMTRwor1fKZH8XKKyiNsuQjJqHi84sh2ZeK
                                        fGmGPEZpDZR+Pk2biQSRpJo9tH29B0fsSE/O
                                        fGDjImgkRhujnMlA/7RA1OilPF0= )
dns0.dnssec.potaroo.net. 86400  IN A    203.50.0.18
                        86400   RRSIG   A 5 4 86400 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        LpbfDJtud6wqXVLnurpxkCuYtiFakQ0HFkIF
                                        qJfs/R9xGwNizeS4f2+dR/rGnwxTDw522qdT
                                        JFIBXbBR9RG9pSEqOCk/ivNSF8dPc7URbI4e
                                        E0RWkgf9fE87x6cd2CHEaOrcgHDXbCZX594R
                                        oWeutR9WohUPovs0aT1fOt2C9Gs= )
                        10800   NSEC    dns1.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        fGmGPEZpDZR+Pk2biQSRpJo9tH29B0fsSE/O
                                        fGDjImgkRhujnMlA/7RA1OilPF0= )
dns0.dnssec.potaroo.net. 86400  IN A    203.50.0.18
                        86400   RRSIG   A 5 4 86400 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        LpbfDJtud6wqXVLnurpxkCuYtiFakQ0HFkIF
                                        qJfs/R9xGwNizeS4f2+dR/rGnwxTDw522qdT
                                        JFIBXbBR9RG9pSEqOCk/ivNSF8dPc7URbI4e
                                        E0RWkgf9fE87x6cd2CHEaOrcgHDXbCZX594R
                                        oWeutR9WohUPovs0aT1fOt2C9Gs= )
                        10800   NSEC    dns1.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        fxw5MRcKkjR6bcRBaD4u/28sOLKZbVVTjYas
                                        1dBcYyx0aW3lIUpvyIsjERU+oEG+g2DUQui+
                                        2LA6PVntaCbKWfezwGkBtZBGKbwUcfNCdEa7
                                        dNKQv3Aki5qGw1EAlkbahKt1FGbQLwO/WI4g
                                        JFmOpYfcmaLtZdhgyWX60KBGIoY= )
dns1.dnssec.potaroo.net. 86400  IN A    203.50.0.6
                        86400   RRSIG   A 5 4 86400 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        C1NTVm64mJDTDpM+aX07OLWhi92G9l5hkiW5
                                        QbBmmITLq1x7QhMpasSPh41PRpa+teyeByFl
                                        /46QGRpVb8IP4KmpbURd1YkPwAJbBBwb2Q+s
                                        dXA6vfz3R/GSa62vSb2aCPfpvAAPkE3Hs66m
                                        DF3DwVONpGuSgAWpn3A3H+1KbQs= )
                        10800   NSEC    sub.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        RfjymANoHG3TQ909fU/lenv3GsIZtEqR6fs7
                                        fa/KJ4o4/OZU7+/VGz3CgUwBOLeMBab9f+Yr
                                        KuFi83KvAt/W4E0nGxeDwgtnkTzUQJpkv7lA
                                        AStqMIrqsZc8FyGJuZPJgU8Fzvn7+Ju7qsPU
                                        Ntwi658ZRKoUl/K7uok6O7HmGSE= )
sub.dnssec.potaroo.net. 86400   IN NS   dns0.dnssec.potaroo.net.
                        86400   IN NS   dns1.dnssec.potaroo.net.
                        10800   NSEC    www.dnssec.potaroo.net. NS RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        XNgHeGnZnmdg8AwHcnsvW6DzZEtnB0n5HVpk
                                        1m2/oFoVgFr7MBuJT1t0beN8p/2zMuLF3Wad
                                        HLmwLX0GqYBE/f+6afIA33aWTrLkuB11cmUj
                                        iEk/4xMKAUgRAN04V6jOvVDnyESXY6g6afd3
                                        J9yhhNvDukG3/8Iq1bUyVlRSKz8= )
www.dnssec.potaroo.net. 86400   IN A    203.50.0.6
                        86400   RRSIG   A 5 4 86400 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        gWXzDRdiVRWxMCseWPQ2oi0QIHQxMZHT+Qj+
                                        nk+tJMW3gvEVH+iP6uLGkwewywey8Ek1bLMe
                                        Uwqlh6z8B35pBBn0hljwO3xO0Ly3ELHvtHUB
                                        Q/2/bDbFaFDaXNA5IQn8I4RGLuaExDKq0dIF
                                        tL/hq9y4rNHg7WTcNw9Q3pRfNUA= )
                        10800   NSEC    dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061008080832 (
                                        20060908080832 3755 dnssec.potaroo.net.
                                        LRLFqls+FF2DqvuPOrnloRe6OclswCG/RL38
                                        X1NLOshkpYjK4GcCsgsoyYCxH2vvmt2va+OU
                                        RqVgL06brBizmmG7raS4kK9yd0bP+91CikWF
                                        HuN8GOLjZ0Sel8CtyOeahtjy7cdqVovPkcje
                                        P1yjDR8cI58wVsdvSCWlaoeCx9k= )
```

The zone file has been sorted into canonical order, NSEC records have been inserted for every label, each RRSet has been signed with the Zone signing key, and the DNSKEY RRset has been signed by both the zone signing key and the key signing key.

Now to change named.conf to reference the signed zone file. This is the new section in the named conf file for the zone server:

```
zone "dnssec.potaroo.net" {
        type master;
        file "master/dnssec.potaroo.net.signed";
};
```

I also need to add the public key signing key of the domain to my trusted key set in the trusted-keys section of named.conf of my local DNS resolver and signal a HUP to the named process to get it to re-read the named.conf file. Here’s my new trusted key for the zone dnssec.potaroo.net on my local name resolver:

```
trusted-keys {

[…]

"dnssec.potaroo.net." 257 3 5 "AQPSOR9BUnuQQ8ien6WibaSsKddzZstW4TEuJrSzezQL79DFqHeOvVuh
  Jr+9JMQmJuQGUjVcXDG1gBRQboiFJ6e+G6sibIKlkzXCLSX7O9YqYtyv
  1AMyEbYWLTwRvKojZSZr2LyKqeKGFqWdoA8a1M6XRuChBlwxMwo5I5fs
  edIyYw==";
};
```

Did it work?

```
# dig +dnssec +multiline DNSKEY dnssec.potaroo.net

; <<>> DiG 9.3.2-P1 <<>> +dnssec +multiline DNSKEY dnssec.potaroo.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27239
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;dnssec.potaroo.net.    IN DNSKEY

;; ANSWER SECTION:
dnssec.potaroo.net.     86400 IN DNSKEY 256 3 5 (
                                AQO8xvbN4hZ8bn926wpM8c9Uqqhqcf45v73k4J/YSu+6
                                o/QsPCKwJoDYxMH3s5ZONJlgLUQscIZZKDYVHPW3Txt5
                                9bHrn739osnQ8ORbOGVTH/Vi//L3BGjZrZr+PWtH2Vb3
                                wIhrujMej2m4E2Mth/XjSDAhYZVWCNhJG0nPH6G6Ww==
                                ) ; key id = 3755
dnssec.potaroo.net.     86400 IN DNSKEY 257 3 5 (
                                AQPSOR9BUnuQQ8ien6WibaSsKddzZstW4TEuJrSzezQL
                                79DFqHeOvVuhJr+9JMQmJuQGUjVcXDG1gBRQboiFJ6e+
                                G6sibIKlkzXCLSX7O9YqYtyv1AMyEbYWLTwRvKojZSZr
                                2LyKqeKGFqWdoA8a1M6XRuChBlwxMwo5I5fsedIyYw==
                                ) ; key id = 29022
dnssec.potaroo.net.     86400 IN RRSIG DNSKEY 5 3 86400 20061008080832 (
                                20060908080832 3755 dnssec.potaroo.net.
                                EMXe2OwX8CNOeAg1iexEMSlGUuApeIB/zW1zpHhZ+I/9
                                YFE2bmmWaj6+jtfMMW8tvjlqdEFH8TOihsMaPhu0nMQn
                                qTrKTNS4Y4DkHqt05N6a3yS1h/ufRfBDn2rA5EquVNGZ
                                M6TRI0iweDSn1HsWy5+FiQcCFubsVJjCyqG/RXo= )
dnssec.potaroo.net.     86400 IN RRSIG DNSKEY 5 3 86400 20061008080832 (
                                20060908080832 29022 dnssec.potaroo.net.
                                plmpAtyiQINPi0RcibIcry9eofJhvm6mkxZSnL5Qb4x/
                                g+DC02kXMhFCsVvNSU9ATAwRIOhYPG85LaC7FdfWdOud
                                5I+AVvVPRB+8aX1scS/8/kQ5AbJuxT3b6ezCEhu2FSuR
                                KN3uskV5Af4N1nBBVmFWd7vXR53Q6KCucWjBvmg= )

;; Query time: 412 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Sep 12 09:44:37 2006
;; MSG SIZE  rcvd: 695
```

The “ad” bit has been set in the flags of the response,, so it looks good.

The name resolver’s debug log also confirms this:

```
validating @0x8247000: dnssec.potaroo.net DNSKEY: starting
validating @0x8247000: dnssec.potaroo.net DNSKEY: attempting positive response validation
validating @0x8247000: dnssec.potaroo.net DNSKEY: verify rdataset: success
validating @0x8247000: dnssec.potaroo.net DNSKEY: signed by trusted key; marking as secure
validator @0x8247000: dns_validator_destroy
```

Lets check for a name that exists within this zone:

```
# dig +dnssec +multiline A www.dnssec.potaroo.net

; <<>> DiG 9.3.2-P1 <<>> +dnssec +multiline A www.dnssec.potaroo.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64058
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.dnssec.potaroo.net.        IN A

;; ANSWER SECTION:
www.dnssec.potaroo.net. 86317 IN A 203.50.0.6
www.dnssec.potaroo.net. 86317 IN RRSIG A 5 4 86400 20061008080832 (
                                20060908080832 3755 dnssec.potaroo.net.
                                gWXzDRdiVRWxMCseWPQ2oi0QIHQxMZHT+Qj+nk+tJMW3
                                gvEVH+iP6uLGkwewywey8Ek1bLMeUwqlh6z8B35pBBn0
                                hljwO3xO0Ly3ELHvtHUBQ/2/bDbFaFDaXNA5IQn8I4RG
                                LuaExDKq0dIFtL/hq9y4rNHg7WTcNw9Q3pRfNUA= )

;; Query time: 75 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Sep 12 13:56:27 2006
;; MSG SIZE  rcvd: 245
```

And check for a name that exists through the wildcard entry

```
# dig +dnssec +multiline A wildcardd.dnssec.potaroo.net

; <<>> DiG 9.3.2-P1 <<>> +dnssec +multiline A wildcardd.dnssec.potaroo.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55385
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 5, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;wildcardd.dnssec.potaroo.net. IN A

;; ANSWER SECTION:
wildcard.dnssec.potaroo.net. 10800 IN A 203.50.0.18
wildcard.dnssec.potaroo.net. 10800 IN RRSIG A 5 3 86400 20061008080832 (
                                20060908080832 3755 dnssec.potaroo.net.
                                UTLlJPY60iaVo8skJKbljkF+DzOZFJiPGSLMEmmzHVlY
                                NeIjpQNK/o5jcIdDv7S4MZ+MJg31MLPXuStBWe8ElfwU
                                4w+eQX38dXP1fPs2Mjz2RyG/dw2krgvVRfQDa27UJVur
                                xDxoQTykEwW7yYzAdA6oVflEkjyTF8O/CxrGVy0= )

;; AUTHORITY SECTION:
sub.dnssec.potaroo.net. 10800 IN NSEC www.dnssec.potaroo.net. NS RRSIG NSEC
sub.dnssec.potaroo.net. 10800 IN RRSIG NSEC 5 4 10800 20061008080832 (
                                20060908080832 3755 dnssec.potaroo.net.
                                XNgHeGnZnmdg8AwHcnsvW6DzZEtnB0n5HVpk1m2/oFoV
                                gFr7MBuJT1t0beN8p/2zMuLF3WadHLmwLX0GqYBE/f+6
                                afIA33aWTrLkuB11cmUjiEk/4xMKAUgRAN04V6jOvVDn
                                yESXY6g6afd3J9yhhNvDukG3/8Iq1bUyVlRSKz8= )
dnssec.potaroo.net.     86294 IN NS dns0.potaroo.net.
dnssec.potaroo.net.     86294 IN NS dns1.potaroo.net.
dnssec.potaroo.net.     86294 IN RRSIG NS 5 3 86400 20061008080832 (
                                20060908080832 3755 dnssec.potaroo.net.
                                p2kKLK4gzlm8nkr4lpXyz4FirWWXtiyXc5X/Ns2NYC3C
                                NYDNIRFHzEI14RZO08R9z4aoQlfOjXidiJZ2BgxzmykV
                                JUaA7AwGirVtr+6wDJrdif9tm7UdYN2powrP9o2lq0DK
                                hwYk8i4Dyjdd9kwt7/x44ZECzEj7w30GfW4uvy8= )

;; Query time: 81 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Sep 12 13:56:50 2006
;; MSG SIZE  rcvd: 693
```

Again this looks good. It appears that as long as the resolver is willing to trust the zone key then I’ve managed to correctly sign a zone and publish the outcomes.

Now I’d like to extend this by signing the delegated zone “sub.dnssec.potaroo.net”. Because this is an immediate descendant of a signed zone then if I set this up correctly then the same trusted key for the parent zone should be able to be used to validate the child zone. The same initial process of zone signing is followed:

1. Generate the zone key and the key-signing keys

```
# dnssec-keygen -r/dev/random -a RSASHA1 -b 1024 -n ZONE sub.dnssec.potaroo.net
Ksub.dnssec.potaroo.net.+005+55625
# dnssec-keygen -r/dev/random -a RSASHA1 -f KSK -b 1024 -n ZONE sub.dnssec.potaroo.net
Ksub.dnssec.potaroo.net.+005+49350
```

2. Add the keys to the zone file, and check the result

```
# xemacs sub.dnssec.potaroo.net &
[add include lines to thezone file for the host and KSK key files]
# named-checkzone -D sub.dnssec.potaroo.net sub.dnssec.potaroo.net
```

```
zone sub.dnssec.potaroo.net/IN: loaded serial 2006091201
sub.dnssec.potaroo.net.                       86400 IN SOA      dns0.dnssec.potaroo.net.
gih.potaroo.net. 2006091201 10800 15 604800 10800
sub.dnssec.potaroo.net.                       86400 IN NS       dns0.dnssec.potaroo.net.
sub.dnssec.potaroo.net.                       86400 IN NS       dns1.dnssec.potaroo.net.
sub.dnssec.potaroo.net.                       86400 IN DNSKEY   256 3 5
AQOoTGeVji8FFqWNqezyxx/C5omVEg98JPzEFSLMbGL32b+Ov9GXxtPM
SGxc+iiJLe2nGuMwgWqLFEUxUeK9P9ZNDR1CurtDKh1KPGQJjf3OJq9B
CffeVaGy9O2KOvLFH9BqVo9L/J5iTcyR3zE0kMoE2rgyYtlUsmUnM1+j usFDrw==  ; key id = 55625
sub.dnssec.potaroo.net.                       86400 IN DNSKEY   257 3 5
AQO4JdCPtVV9IZ87bJHA8ZYOGZZNr5uI9Nc7/Sh1toKIwQHh3DWLp3q/
yAto8m80wtTb7nZr+Jvem9elfWRy6Rqqc/bETiAYOIU9f+qdE87KEipw
RnRIMSCjmcJBhEYbJhWmONBHhT8IdwqtwVFuIAsmIXgI5siG5YAKbl+X XHAOSQ==  ; key id = 49350
another.sub.dnssec.potaroo.net.               86400 IN A        203.50.0.18
example.sub.dnssec.potaroo.net.               86400 IN A        203.50.0.6
www.sub.dnssec.potaroo.net.                   86400 IN A        203.50.0.6
OK
```

3. Bump the SOA value and Sign the child zone

I'm not sure why I put the "-d ." command option to the dnssec-signzone command here. I thought it has something to do with generating the DS and keyset files that I need to pass to the parent zone, but the man page for this command seems to suggest otherwise. Indeed the man page for this command is sufficiently unclear on the entire topic of dsset and keyset files as to be entirely useless to me. I had the distinct impression that I was walking in the dark at this point in time.

```
# /usr/local/sbin/dnssec-signzone -r /dev/random  -d . -o sub.dnssec.potaroo.net -k
Ksub.dnssec.potaroo.net.+005+49350 sub.dnssec.potaroo.net Ksub.dnssec.potaroo.net.+005+55625.key
sub.dnssec.potaroo.net.signed
# cat sub.dnssec.potaroo.net.signed
; File written on Tue Sep 12 14:53:06 2006
; dnssec_signzone version 9.3.2
sub.dnssec.potaroo.net. 86400   IN SOA  dns0.dnssec.potaroo.net. gih.potaroo.net. (
                                        2006091201 ; serial
                                        10800      ; refresh (3 hours)
                                        15         ; retry (15 seconds)
                                        604800     ; expire (1 week)
                                        10800      ; minimum (3 hours)
                                        )
                        86400   RRSIG   SOA 5 4 86400 20061012035306 (
                                        20060912035306 55625 sub.dnssec.potaroo.net.
                                        Qh6jxu5LXSfD4OpWTM0ilU2rjN74GJNkvdJ1
                                        GOlu+jywnNDKAwAJ4oRAfmp64/P+KZRWHUnM
                                        qMpbkWc+12yFANXPGWOxhRFG0dOXUV5iYZSh
                                        rTzS134CzksmDDQoTljQf68vD6Oowm5vBGTL
                                        3PmJDDMIVykk9MPJ4xZHGhMqZUs= )
                        86400   NS      dns0.dnssec.potaroo.net.
                        86400   NS      dns1.dnssec.potaroo.net.
                        86400   RRSIG   NS 5 4 86400 20061012035306 (
                                        20060912035306 55625 sub.dnssec.potaroo.net.
                                        B0lEobJvCHagIQR263PO6FnYnHyGP7npbKzl
                                        0ff6uHn40ZYXHws8USEebC+l6m/dCbBxPfiR
                                        Y0lP4p9G/3cdUG8TAWKoNLIhxZ7JnJEXmmO1
                                        M32PhZcBjEwtSMxxOp7LZ5+aBINvS6JmWnY1
                                        JXkPvSENLMpDTtNwz/BQJeE10hI= )
                        10800   NSEC    another.sub.dnssec.potaroo.net. NS SOA RRSIG NSEC DNSKEY
                        10800   RRSIG   NSEC 5 4 10800 20061012035306 (
                                        20060912035306 55625 sub.dnssec.potaroo.net.
                                        JqKFuVi5CTl1iSKXZxep1HfLNc4BUq4hj14Z
                                        ggD37Jo5VX8FWj4piRAnQSWl11G0p8G2Fl+b
                                        YTqX4mn0kM1hIYsyqH2O8wnJcpeGGaYfho15
                                        7HiwLOCNHeL9CSSpNbFR5J4AORm46b16M0Q/
                                        tmcZHdzNoXKENUvix6rUN8kYCJU= )
                        86400   DNSKEY  256 3 5 (
                                        AQOoTGeVji8FFqWNqezyxx/C5omVEg98JPzE
                                        FSLMbGL32b+Ov9GXxtPMSGxc+iiJLe2nGuMw
                                        gWqLFEUxUeK9P9ZNDR1CurtDKh1KPGQJjf3O
                                        Jq9BCffeVaGy9O2KOvLFH9BqVo9L/J5iTcyR
                                        3zE0kMoE2rgyYtlUsmUnM1+jusFDrw==
                                        ) ; key id = 55625
                        86400   DNSKEY  257 3 5 (
                                        AQO4JdCPtVV9IZ87bJHA8ZYOGZZNr5uI9Nc7
                                        /Sh1toKIwQHh3DWLp3q/yAto8m80wtTb7nZr
                                        +Jvem9elfWRy6Rqqc/bETiAYOIU9f+qdE87K
                                        EipwRnRIMSCjmcJBhEYbJhWmONBHhT8Idwqt
                                        wVFuIAsmIXgI5siG5YAKbl+XXHAOSQ==
                                        ) ; key id = 49350
                        86400   RRSIG   DNSKEY 5 4 86400 20061012035306 (
                                        20060912035306 49350 sub.dnssec.potaroo.net.
                                        D5svOlt6E8v/IakXQyVdXqQXWMZZvY1YqiHA
                                        2EeEto8TNUgHs1vSme3rW+88YjfxwXk2GfE1
                                        rO1RaabtCi9OcNk60cJ98FSySjIVkVpRW2xB
                                        k2S4SBWz0uwBq2VbFhQc1AkLOkZR97ef4GkC
                                        hchvuf656sPYun3QZd285wOs0J8= )
                        86400   RRSIG   DNSKEY 5 4 86400 20061012035306 (
                                        20060912035306 55625 sub.dnssec.potaroo.net.
                                        UJyhU+eTh/cSJ/4dsdvBjbT0c4MED3Z+mU1b
                                        kmhuChtKboLEWeVhDXFSBk7S3ugemZJlFQbL
                                        lwOF4ZfemZq+V/PLnD3+u3wMywn5AKJ2cjoY
                                        W2CWk25qyaZef23YtMI1Guq3qs0d9KCb9NCW
                                        wra3YiOBx8jTjr1VFMnpPzmxN58= )
another.sub.dnssec.potaroo.net. 86400 IN A 203.50.0.18
                        86400   RRSIG   A 5 5 86400 20061012035306 (
                                        20060912035306 55625 sub.dnssec.potaroo.net.
                                        ZOGW6QvuU03lS4cmnIjlNWyXUQ5/LcnJw1bU
                                        hje1MM+o1yuQMMUxpWPJSfe7R7mJ6u8k3LTU
                                        P+wBi7hF4S07xDO8VLD5XmN6capRfUMkasKO
                                        XOHg+2+mUSvOOEY+0vl4/7R6vZM3XjseoDRK
                                        l/bgeoawKEf5dIdB+Xlm425tu9I= )
                        10800   NSEC    example.sub.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 5 10800 20061012035306 (
                                        20060912035306 55625 sub.dnssec.potaroo.net.
                                        O6m+XooOVuFIHVcLqd3Vqy7WeDos2CV2xShr
                                        YPCmBoUKF774BnYrKMPe+HO5p65kiyTh3w9L
                                        z1Iayzh6SN+HVrrf6vMYgIVlxyqScnpMdzMT
                                        vS5VP2bHErr6WudXUdCb6XZSACB2OCT5K2MW
                                        A3/sYQoFQyGMamp5aETrReAJwxQ= )
example.sub.dnssec.potaroo.net. 86400 IN A 203.50.0.6
                        86400   RRSIG   A 5 5 86400 20061012035306 (
                                        20060912035306 55625 sub.dnssec.potaroo.net.
                                        i/cotH8MpAjIiEiOjwMsvN0T81aQQnSkuxWO
                                        boLRgtDpF3Si6oGMM2AhHLgnJDepvSGBqqEw
                                        YqlagfhebF/joC7F8K3GxUl/dujg1eosTY7w
                                        eL3adFRKBMGqkkdw7OrY417f8VvH2hg2wG+P
                                        KM0NLA3DmxQ16jqd3f9cc9Tz2Bg= )
                        10800   NSEC    www.sub.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 5 10800 20061012035306 (
                                        20060912035306 55625 sub.dnssec.potaroo.net.
                                        OHxc6vMcbm4Its2BMlbMs0QGLu6KdVroqCSA
                                        F9iaZsOyvbnzRBDr6tyC3ANJh8Ld48CA9uPh
                                        8HWI13iU5uiDcoELYFUBZl8vPakKOenDB52b
                                        XBKbWCub51nvjs4PMNS4MnsJTc82ePeHhOJK
                                        5cKW3uodx2TmaAaOpf8V2kugaMA= )
www.sub.dnssec.potaroo.net. 86400 IN A  203.50.0.6
                        86400   RRSIG   A 5 5 86400 20061012035306 (
                                        20060912035306 55625 sub.dnssec.potaroo.net.
                                        PyejoxvnwdMdmOBAaMWxg649TQLTFTL9JBwi
                                        hXzxB1F3I5YC1gHrNRzYUJbyaZOFB3JM9nY+
                                        zOI+TF25hm3tMXlEPjN6eAYk7CHh5T8XHJmv
                                        rgI2Ermxs7+DFbLOB1CPOsSRS8m8/L7JlvCP
                                        8l0ZqKTbECUbqY8bTFwznhD/wvg= )
                        10800   NSEC    sub.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 5 10800 20061012035306 (
                                        20060912035306 55625 sub.dnssec.potaroo.net.
                                        k7uDto4S6XVqafvO1BNVnnU3EG/Fsz42HoYf
                                        atqbL5a71Syu67ATQE5NOimWB7LJIBn74Go/
                                        dQGc526tq2xCCs3L/zqk3VgkyIoEE8tiGFaE
                                        8OXhATFnnpAjGGiApLk9d+uWBaZ7cg3cR+V4
                                        kKv51mjg8h+1bS5aydKgmq7QZdE= )
```

4. Serve the signed child zone.

```
# cd /etc/namedb
# xemacs named.conf
[reference sub.dnssec.potaroo.net.signed as the zone file for this domain]
# ps axuw | grep named | grep -v grep
root  68795  0.0  0.1  4328  3332  ??  Ss   Sat10AM   0:00.77 /usr/local/sbin/named -c
/etc/namedb/named.conf -d 10
# kill -s HUP 68795
```

5. Check that the new signed zone has been loaded by the server

```
# dig SOA sub.dnssec.potaroo.net @127.0.0.1

; <<>> DiG 9.3.2 <<>> SOA sub.dnssec.potaroo.net @127.0.0.1
; (1 server found)
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19161
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;sub.dnssec.potaroo.net.                IN      SOA

;; ANSWER SECTION:
sub.dnssec.potaroo.net. 86400   IN      SOA     dns0.dnssec.potaroo.net. gih.potaroo.net. 2006091201
10800 15 604800 10800

;; AUTHORITY SECTION:
sub.dnssec.potaroo.net. 86400   IN      NS      dns0.dnssec.potaroo.net.
sub.dnssec.potaroo.net. 86400   IN      NS      dns1.dnssec.potaroo.net.

;; ADDITIONAL SECTION:
dns0.dnssec.potaroo.net. 86400  IN      A       203.50.0.18
dns1.dnssec.potaroo.net. 86400  IN      A       203.50.0.6

;; Query time: 1 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Sep 12 14:59:09 2006
;; MSG SIZE  rcvd: 150
```

6. Now I need to generate a DS Resource Record for the child zone that can be placed into the parent zone

A side effect of the dnssec-signzone call in step 3 was the generation of the keyset and dsset files

```
# ls -al *set-sub*
-rw-r--r--  1 root  wheel   81 Sep 13 09:20 dsset-sub.dnssec.potaroo.net.
-rw-r--r--  1 root  wheel  289 Sep 13 09:20 keyset-sub.dnssec.potaroo.net.
# cat keyset-sub.dnssec.potaroo.net.
$ORIGIN .
sub.dnssec.potaroo.net  10800   IN DNSKEY 257 3 5 (
                                        AQO4JdCPtVV9IZ87bJHA8ZYOGZZNr5uI9Nc7
                                        /Sh1toKIwQHh3DWLp3q/yAto8m80wtTb7nZr
                                        +Jvem9elfWRy6Rqqc/bETiAYOIU9f+qdE87K
                                        EipwRnRIMSCjmcJBhEYbJhWmONBHhT8Idwqt
                                        wVFuIAsmIXgI5siG5YAKbl+XXHAOSQ==
                                        ) ; key id = 49350
# cat dsset-sub.dnssec.potaroo.net.
sub.dnssec.potaroo.net. IN DS 49350 5 1 075734C803444B198C0681A8FD9A9E4378E4F4B2
```

I’m not sure which of these should be passed to the parent zone admin, so I’ll hurl both files across the fence and leave it to the parent to figure out what to do!

7. Now to get the parent zone key to sign the dsset record for the child domain

I’ve added the DS RR to the parent zone file. Something feels wrong here, and I’m not sure why I’m doing this. The RIPE tutorial document gets pretty unclear at this stage, and the man page for the dnssec-signzone command appears to suggest that I put the keyset file in a special directory and get the dnssec-signzone command to generate the ds-set files via the –d command option referencing the location of the key-set files for the child zone. This approach did not work for me, and after a series of experiments I found that if I took the child zone’s ds-set file contents and inserted this DS record directly into the parent zone file for the subdomain, then things appear to work. I couldn’t find any documentation that confirmed that this was the ‘correct’ way to get the child’s key information into the parent zone, so I probably did something wrong here. Did I mention already that I found the DNSSEC dnssec-signzone utility documentation terse and less than helpful?

```
# xemacs dnssec.potaroo.net
[add dsset-sub.dnssec.potaroo.net]
# cat dnssec.potaroo.net
$TTL 86400

$ORIGIN              dnssec.potaroo.net.

@               IN      SOA     dns0.potaroo.net. gih.potaroo.net. (
                2006091302 ; Serial
                3h   ; Refresh
                15   ; Retry
                1w   ; Expire
                3h ) ; Minimum
;
;
;
; Name servers
;
                IN      NS      dns0.potaroo.net.
                IN      NS      dns1.potaroo.net.
;
; subdomains
;
sub                  IN       NS       dns0.dnssec.potaroo.net.
           IN       NS       dns1.dnssec.potaroo.net.
           IN       DS       49350 5 1 075734C803444B198C0681A8FD9A9E4378E4F4B2
;
; zone A records
;
www                  IN       A        203.50.0.6
bgp                  IN       A        203.50.0.159
bgp2                 IN       A        203.50.0.33
dns0                 IN       A        203.50.0.18
dns1                 IN       A        203.50.0.6
;
; wildcard
;
*          IN       A        203.50.0.18

$include Kdnssec.potaroo.net.+005+03755.key      ; zone signing key
$include Kdnssec.potaroo.net.+005+29022.key      ; key singing key
```

8. Now re-sign this zone file

Not forgetting to bump the SOA value beforehand, of course.

```
# /usr/local/sbin/dnssec-signzone -r /dev/random -o dnssec.potaroo.net. -k
Kdnssec.potaroo.net.+005+29022 dnssec.potaroo.net Kdnssec.potaroo.net.+005+03755.key
dnssec.potaroo.net.signed
# cat dnssec.potaroo.net.signed
; File written on Wed Sep 13 09:34:58 2006
; dnssec_signzone version 9.3.2
dnssec.potaroo.net.     86400   IN SOA  dns0.potaroo.net. gih.potaroo.net. (
                                        2006091302 ; serial
                                        10800      ; refresh (3 hours)
                                        15         ; retry (15 seconds)
                                        604800     ; expire (1 week)
                                        10800      ; minimum (3 hours)
                                        )
                        86400   RRSIG   SOA 5 3 86400 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        tpRo4CwylVHGBctDcZyhzc+Gqqca2avlbE9j
                                        Pfn3JwSti/eyHxEYB7EL0J18bDXa7xfBoq59
                                        O0TizV4K1p1se6qRDxUEp6FQ32R6DMDkOxU3
                                        YdMMkScyOR2+nZ/VaslPvlRLdhfrBhN53HmR
                                        L+VRuT+VGYDYFqq2AJco4NPV8Go= )
                        86400   NS      dns0.potaroo.net.
                        86400   NS      dns1.potaroo.net.
                        86400   RRSIG   NS 5 3 86400 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        qCqS29jXtzVr9GYPrLszOwBf0vsSWAcmWSpV
                                        NyrL1LFf6VgF4/trYOXdQYssROn55ORXqKYX
                                        mtIyijG/l1RU8Hi0fkjilPrxOdw6+DM1bcMS
                                        6xIPN8GPLx6aOh4k6FE83xg5jKRbei7xzyVy
                                        hM0OrUxkDwlMhXSe1CIO7bcsUWo= )
                        10800   NSEC    *.dnssec.potaroo.net. NS SOA RRSIG NSEC DNSKEY
                        10800   RRSIG   NSEC 5 3 10800 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        XFIQUy+kU2fg6h1dKu1CCIv+B0UjudRzsMSh
                                        +VwVJv0bsli4rdWarfpejgpoZ57YfQhrzFWb
                                        AB+OEmDrhAnkRjstB2XzjMt2UDHu9AgXMJaB
                                        XONfDtBvNWBquZ5VHdZd4iPH0aCcA577Gul4
                                        AdoFB9Ec9NZ5OoE6po/h0t3NGww= )
                        86400   DNSKEY  256 3 5 (
                                        AQO8xvbN4hZ8bn926wpM8c9Uqqhqcf45v73k
                                        4J/YSu+6o/QsPCKwJoDYxMH3s5Z0NJlgLUQs
                                        cIZZKDYVHPW3Txt59bHrn739osnQ80RbOGVT
                                        H/Vi//L3BGjZrZr+PWtH2Vb3wIhrujMej2m4
                                        E2Mth/XjSDAhYZVWCNhJG0nPH6G6Ww==
                                        ) ; key id = 3755
                        86400   DNSKEY  257 3 5 (
                                        AQPSOR9BUnuQQ8ien6WibaSsKddzZstW4TEu
                                        JrSzezQL79DFqHeOvVuhJr+9JMQmJuQGUjVc
                                        XDG1gBRQboiFJ6e+G6sibIKlkzXCLSX7O9Yq
                                        Ytyv1AMyEbYWLTwRvKojZSZr2LyKqeKGFqWd
                                        oA8a1M6XRuChBlwxMwo5I5fsedIyYw==
                                        ) ; key id = 29022
                        86400   RRSIG   DNSKEY 5 3 86400 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        AEm4XejIj2PwlZZZUSrbaq051YOT9Q1UbWlH
                                        835QmhWsdQl7NW2rSp6CVnsTGpmLK4XIA64b
                                        d72h6VGjlFlzsuzqTBfu0/GcxY5FV+fSs0vQ
                                        EiwoDfK41CqViQD9ogzH5P6uXyHbvk5FHDgN
                                        Qpfsh1UWe0h7oU6VcmSx/RVpSNE= )
                        86400   RRSIG   DNSKEY 5 3 86400 20061012223458 (
                                        20060912223458 29022 dnssec.potaroo.net.
                                        xDH4b6m8NZcPghrUZF2L12T8ZH0ymAso5nVB
                                        h42ZmaF5eK7Wqeq5BBYeIEf31FwqF3zSxujC
                                        BsfItUGVxYyOeAru44pCMw8WR1Fxk8o/D7B5
                                        yp9vZdpI6lDIdK+iMO6h0ce3zQF5dQ81T/XS
                                        hwcZQsQPgcHUN/ys3rBcfrkluNY= )
*.dnssec.potaroo.net.   86400   IN A    203.50.0.18
                        86400   RRSIG   A 5 3 86400 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        rxxOXTio72b5HY5ad7NI6/JwxYIeP6P/sjoK
                                        PpTlA28uNA2E6RUryujGRF0/P2EHap4r+eqI
                                        PkOclbaOtXs83/xTjnGEzsjWQNqz0e8glKUP
                                        c/J6+/sTAM1ja7zLDr3fRqdunH2WYECjPWY/
                                        6Av2Ow6erln5vWbsIyS+/2uxcvw= )
                        10800   NSEC    bgp.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 3 10800 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        R5vkZvePoE/ISt3/PUeZYMPDnseGxcidxahu
                                        5QyBEd1yaSzRJLNHeA1VMxgvchcH7L1tR8wY
                                        RTLnR173IcjJfhTtqQUy89JvjdQLcVZZ+eRg
                                        26t2fbvpzz4/kYDX67uOjSScFXmCyIOtqgTS
                                        Rs+fgiOhA7RIm9C1ZNzE2TqUij0= )
bgp.dnssec.potaroo.net. 86400   IN A    203.50.0.159
                        86400   RRSIG   A 5 4 86400 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        plsnUAtTuRILC4A3TwdSx9TeGG2qutZL0hay
                                        Wc07OF9AUZ0Hepiy9uhG08SBh0eGyJj9dtsp
                                        iLhfPgB1WAax9WjA4emAlwfbxoCj6DhGviQB
                                        x9krpVQAZmjaDGyQUhC/yI1stU1mEMezWktw
                                        23+BLSAla6RyAc+5p0/iiQxOBJ0= )
                        10800   NSEC    bgp2.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        axWY/ImL+jT7d5Px8QqNzMqMxh+t/xnmoN05
                                        n+oaAuHGfHBxbJhCxNGSb9eWfYInQMOgg+Ta
                                        /a85LZp44c6o2xo46ElBuuYYt/V5kud/p4UN
                                        qsKGuvDwVPT4aSkkQREDOULB0bR4XgIP8Ttj
                                        4mbFPNh+L6/GGt/NpMpOK8Vm/n0= )
bgp2.dnssec.potaroo.net. 86400  IN A    203.50.0.33
                        86400   RRSIG   A 5 4 86400 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        liWq7aC32VT+wlTxU9CMm0g5RjgFDRersSYg
                                        7DryptyYs7SHCl+KnOIkslzpalivWC/HXTs1
                                        FiyR2bAKJz3qTs+VRaCp8o3Zcf65W1EZ+MNi
                                        WxEzWIIuu0e3WY3ofF4KY6AHa+Yu4trGKosn
                                        95GGKxAxYzFKXXnifcGLLzy2W6k= )
                        10800   NSEC    dns0.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        YwyqvugvmxuhCD9ldzLooUR5Q9YFbCIrsQ3
                                        96BsPJoQSu1xztu8ptNvvxcK1NjVle+cxJg/
                                        /BeWA/4B6sdRcFwrDl1LWWHrZDOfMFSCOWnW
                                        jkTqCVIrXzvD89c0cgTEtNqYxap/j4zDTnmv
                                        jPB2bXT3mnciEdNaRSEpuQHU7L8= )
dns0.dnssec.potaroo.net. 86400  IN A    203.50.0.18
                        86400   RRSIG   A 5 4 86400 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        CudPXRmsA1/bgGC3XrYkjqpATG5bYxcLSc3q
                                        LkWGU27wIW49UKL7J8AdC6bENSrrymoOFeyF
                                        QzBHxnvck+6CoedsVNUQ2wEbjSshMvl3JHV4
                                        Ou09oepvBh+QKkh/4VLbsvDa+y7wq28jnUvv
                                        lP33xU5g1TWIUYxO3sPkuo/jcrQ= )
                        10800   NSEC    dns1.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        s8bI/U+CvSbfmnxDd1X+MbH3eNtWTvSIQLWG
                                        Y3ZA5a2zMWmI1NTaCJ7rhUzkS4sXQuVHpBVI
                                        cTsueQK5TUwN7Y55WOi0FqCiRgeFd3Z81RDm
                                        pHOt+t+3U1BCY9Qmpx80+RAvrVQSz9H0muHY
                                        cA0ss5b1Qn4CjHlSeJ1vWGqQBN4= )
```

```
dns1.dnssec.potaroo.net. 86400  IN A    203.50.0.6
                        86400   RRSIG   A 5 4 86400 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        ZnBFNuYbuegNnJHZckGvoHM8dOVal/RpXMlr
                                        V2B3lj8fiwREEmkVKNXqRjb5uVtKRPEVL/cl
                                        YZDGtlTpoE9RNh+kuoqFDa8W5g5UjP/TpmLC
                                        afz1QUCPIetCOOrsqJUO+KQXnbxzgOlrQCMQ
                                        UuPgFeBST5qweHGZK3GLfuK9d2s= )
                        10800   NSEC    sub.dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        N5QJ2KguZSg0IZf8BCJYenZbInj9ruODD5yq
                                        UUh1gv1DxC+N2d1zKV6Q4QMGFQdBZYub5avi
                                        F28cTKVJRoYHAW3nL7OaoqdYnSoEpmIwL4B8
                                        v4oHjeW0fNgyLu0X9YRTUEVUHwS4IlnZvgDZ
                                        5Eoo1h4vNDKcDvfjRShubz3KQxc= )
sub.dnssec.potaroo.net. 86400   IN NS   dns0.dnssec.potaroo.net.
                        86400   IN NS   dns1.dnssec.potaroo.net.
                        86400   DS      49350 5 1 (
                                        075734C803444B198C0681A8FD9A9E4378E4
                                        F4B2 )
                        86400   RRSIG   DS 5 4 86400 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        TwIT6IfYScIaaxbJ5+/1OJThRWXOz+Gyjj7K
                                        Y4dJIoBH7oVylPH3gUedxhQBubuxhj1vZiQz
                                        55IJxf1/m6ceS6d7AVTD8aM/P+0geu89v5qN
                                        aqJ4s/JB1EeHmpNqeuPOwZ0qX1AAIrnn5lFc
                                        9aYVGvbri79O8eUVEH4tLexY3xU= )
                        10800   NSEC    www.dnssec.potaroo.net. NS DS RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        KCRDoDQn89346cB/08W1bx9zYjS9p2aaUB3W
                                        BLcOcop6sKP+pHvtDlt+FceGCkg8IDEHXFiW
                                        swdcFGK6sO7il9wmWevpw0hiUMM4oGhTxw4A
                                        OVU3ZIp0nkrvljOFOLa9c6z+vnaLWH2bpx+X
                                        GaAEvIAaHq+iR9mqraC78DBCx6c= )
www.dnssec.potaroo.net. 86400   IN A    203.50.0.6
                        86400   RRSIG   A 5 4 86400 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        QAOQtsaGekXCg6Kb4PxmF8WyU9E0KVxAm+tT
                                        Lsy0RnmjSjokvAEHZeIdykKk7M5+TtdKVKHI
                                        qR20MaKXASRBPMROS/kRsq0ikZTA6zPj70EU
                                        eEZ0ApNDx4TbYZz7sgE9R2KsIwN2DJ9YK4FW
                                        6abrfJ2Z0x4tFxrZqsbTEaDWL34= )
                        10800   NSEC    dnssec.potaroo.net. A RRSIG NSEC
                        10800   RRSIG   NSEC 5 4 10800 20061012223458 (
                                        20060912223458 3755 dnssec.potaroo.net.
                                        XhDJAbsgaKZrCGxlu0xBeXSZMDtgpL5HAO8d
                                        nf8398zWxgasHJxIlR4ydFateJiPNuWkawev
                                        V0X2bO9fu+pUpihbDK1XReEQpG0ExLxm/YGJ
                                        MuYTTfvWfd748CVZso5cEgahFYU++n5nMvBj
                                        EgwTtiZRoJSHDYsjYLTxSwNnOKs= )
```

Note in that now-voluminous zone file that the parent has now signed the child domain’s DS record.

9. Now signal the server’s named process to re-read the config file

This will reload the zone information for the zone dnssec.potaroo.net in order to serve the updated zone file.

Phew! I think we are done! Over on the local DNS resolver, which is configured with the trust key of the parent zone (dnssec.potaroo.net) but not the child zone (sub.dnssec.potaroo.net), then the above steps should allow the resolver to validate answers from the child zone.

Lets check. The parent zone responses can be validated:

```
# dig +dnssec +multiline SOA dnssec.potaroo.net

; <<>> DiG 9.3.2-P1 <<>> +dnssec +multiline SOA dnssec.potaroo.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47873
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;dnssec.potaroo.net.    IN SOA

;; ANSWER SECTION:
dnssec.potaroo.net.     86400 IN SOA dns0.potaroo.net. gih.potaroo.net. (
                                2006091302 ; serial
                                10800      ; refresh (3 hours)
                                15         ; retry (15 seconds)
                                604800     ; expire (1 week)
                                10800      ; minimum (3 hours)
                                )
dnssec.potaroo.net.     86400 IN RRSIG SOA 5 3 86400 20061012223458 (
                                20060912223458 3755 dnssec.potaroo.net.
                                tpRo4CwylVHGBctDcZyhzc+Gqqca2avlbE9jPfn3JwSt
                                i/eyHxEYB7EL0J18bDXa7xfBoq59O0TizV4K1p1se6qR
                                DxUEp6FQ32R6DMDkOxU3YdMMkScyOR2+nZ/VaslPvlRL
                                dhfrBhN53HmRL+VRuT+VGYDYFqq2AJco4NPV8Go= )

;; Query time: 391 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Sep 13 09:34:09 2006
;; MSG SIZE  rcvd: 270
```

And the child zone responses can be validated:

```
# dig +dnssec +multiline SOA sub.dnssec.potaroo.net

; <<>> DiG 9.3.2-P1 <<>> +dnssec +multiline SOA sub.dnssec.potaroo.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17696
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;sub.dnssec.potaroo.net.        IN SOA

;; ANSWER SECTION:
sub.dnssec.potaroo.net. 86400 IN SOA dns0.dnssec.potaroo.net. gih.potaroo.net. (
                                2006091201 ; serial
                                10800      ; refresh (3 hours)
                                15         ; retry (15 seconds)
                                604800     ; expire (1 week)
                                10800      ; minimum (3 hours)
                                )
sub.dnssec.potaroo.net. 86400 IN RRSIG SOA 5 4 86400 20061012035306 (
                                20060912035306 55625 sub.dnssec.potaroo.net.
                                Qh6jxu5LXSfD4OpWTM0ilU2rjN74GJNkvdJ1GOlu+jyw
                                nNDKAwAJ4oRAfmp64/P+KZRWHUnMqMpbkWc+12yFANXP
                                GWOxhRFG0dOXUV5iYZShrTzS134CzksmDDQoTljQf68v
                                D6Oowm5vBGTL3PmJDDMIVykk9MPJ4xZHGhMqZUs= )

;; Query time: 124 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Sep 13 09:34:38 2006
;; MSG SIZE  rcvd: 278
```

The named dnssec log on the resolved also indicates that this is functioning correctly:

```
validating @0x81cc000: dnssec.potaroo.net SOA: starting
validating @0x81cc000: dnssec.potaroo.net SOA: attempting positive response validation
validating @0x81cc000: dnssec.potaroo.net SOA: get_key: creating fetch for dnssec.potaroo.net DNSKEY
  validating @0x81cc800: dnssec.potaroo.net DNSKEY: starting
  validating @0x81cc800: dnssec.potaroo.net DNSKEY: attempting positive response validation
  validating @0x81cc800: dnssec.potaroo.net DNSKEY: verify rdataset: success
  validating @0x81cc800: dnssec.potaroo.net DNSKEY: signed by trusted key; marking as secure
  validator @0x81cc800: dns_validator_destroy
  validating @0x81cc000: dnssec.potaroo.net SOA: in fetch_callback_validator
  validating @0x81cc000: dnssec.potaroo.net SOA: keyset with trust 7
  validating @0x81cc000: dnssec.potaroo.net SOA: resuming validate
  validating @0x81cc000: dnssec.potaroo.net SOA: verify rdataset: success
  validating @0x81cc000: dnssec.potaroo.net SOA: marking as secure
  validator @0x81cc000: dns_validator_destroy

validating @0x8249000: sub.dnssec.potaroo.net SOA: starting
  validating @0x8249000: sub.dnssec.potaroo.net SOA: attempting positive response validation
validating @0x8249000: sub.dnssec.potaroo.net SOA: get_key: creating fetch for
sub.dnssec.potaroo.net DNSKEY
```

```
  validating @0x8249800: sub.dnssec.potaroo.net DNSKEY: starting
  validating @0x8249800: sub.dnssec.potaroo.net DNSKEY: attempting positive response validation
  validating @0x8249800: sub.dnssec.potaroo.net DNSKEY: validatezonekey: creating fetch for
sub.dnssec.potaroo.net DS
  validating @0x8251000: sub.dnssec.potaroo.net DS: starting
  validating @0x8251000: sub.dnssec.potaroo.net DS: attempting positive response validation
  validating @0x8251000: sub.dnssec.potaroo.net DS: keyset with trust 7
  validating @0x8251000: sub.dnssec.potaroo.net DS: verify rdataset: success
  validating @0x8251000: sub.dnssec.potaroo.net DS: marking as secure
  validator @0x8251000: dns_validator_destroy
  validating @0x8249800: sub.dnssec.potaroo.net DNSKEY: in dsfetched
  validating @0x8249800: sub.dnssec.potaroo.net DNSKEY: dsset with trust 7
  validating @0x8249800: sub.dnssec.potaroo.net DNSKEY: verify rdataset: success
  validating @0x8249800: sub.dnssec.potaroo.net DNSKEY: marking as secure
  validator @0x8249800: dns_validator_destroy
  validating @0x8249000: sub.dnssec.potaroo.net SOA: in fetch_callback_validator
  validating @0x8249000: sub.dnssec.potaroo.net SOA: keyset with trust 7
  validating @0x8249000: sub.dnssec.potaroo.net SOA: resuming validate
  validating @0x8249000: sub.dnssec.potaroo.net SOA: verify rdataset: success
  validating @0x8249000: sub.dnssec.potaroo.net SOA: marking as secure
  validator @0x8249000: dns_validator_destroy
```

Success! Of a sort…

Step 5 – Evaluating the Outcome

After working with this, off and on I admit, for more than two weeks, I've been able to partially achieve the initial objective. I've configured a DNSSEC-aware local resolver. I've failed in creating a dnssec-aware application interface that would support an analog of the gethostbyname() call, with an additional functional module that would perform a DNSSEC validation of the outcome and raise an alert if the validation pass failed.

The issue here is not the task of configuring the Bind software to perform DNSSEC validation (that's the easy bit!), but that of understanding how to configure trust in the resolver. It appears that the original vision of DNSSEC was that of a self-contained system that started with explicit configuration of the trust anchor of the DNS root, and then use DS chains to have the parent sign the keys of all of its delegated children, and so on. Authentication of a DNS response in such an environment would be that the response matches the signed data in the RRSIG record, and that the key validates against the zone's parent, and so on right up to the root's trust anchor. Like the named.boot file that loads the DNS resolver with an initial seed set of bindings for the DNS root servers, the original vision of DNSSEC appeared to encompass a similar single top level injection of trust, from which all other DNSSEC trust relationships could be derived. The current reality of DNSSEC falls far short of this all-encompassing vision, and the task of funding out which zones have been signed, and establishing in a secure manner the keys of each of these DNSSEC "islands" appears to be one where the effort far outweighs the resultant benefit. Even the efforts with the DLV lookaside trust validation tool have been unsatisfying for me, and I've been unable to use one instance of this tool to validate any DNS domains.

I can't see any applications that use DNSSEC-enabled queries into the DNS, nor can I see any realistic mechanism to load my resolver with trust keys that reflect the partial deployment of DNSSEC today. So, from the perspective of a DNS resolver client I must admit that can't see how DNSSEC is relevant today beyond the exercise of technology-proving, leading to the conclusion that DNSSEC could work, and possibly work reasonably well, as long as all DNS zone servers used it!

So how did the zone signing and serving exercise go? My efforts in zone signing appears to have been successful, despite the poor standard of the tool documentation at present. This part of the exercise has taken me over a week to complete, and the basic reason lies in the relatively incomplete state of documentation of the toolset. The changes in the tool interface across Bind versions has not helped, and there are still a number of out-of-date online guides that reference tools that are not part of the Bind9.3.2 toolset that are picked up by the more popular online search engines. The lack of precise step-by-step instructions that show how to publish a signed zone is also a weakness, and I certainly spent some time in a trial-and-error search as to what sequence of actions would produce a linkage between a parent and child zone key set. Yes, I could've requested assistance at any time, and I'd guess that with assistance from someone who had been through this in the past this entire effort could've been accomplished in a few minutes rather than the days it took me. The problem is not in the complexity of the procedures. The problem as I see it is one of scant documentation that does not take the perspective of a potential user of the technology.

I have taken an easy route through some of this work, such as using NSEC records rather than experimentation with NSEC3 records. I have not performed key rollover of the Zone signing key nor the Key-signing key. I have not attempted to perform trusted communication between the master zone server and its secondaries (using TSIG), nor did I attempt to secure the communication between the parent and child zones in order to pass the key set. Even so the experience has left me feeling that DNSSEC, as a packaged technology for mass consumption, is still very rough at the edges.

Further Reading

DNSSEC HOW TO, A Tutorial in Disguise, Olaf Kolkman, RIPE NCC, April 2005,
http://www.ripe.net/disi/dnssec_howto/dnssec_howto.pdf
Overall I found this to be a useful guide to DNSSEC deployment, although I have to admit that Chapter 3 left me slightly dazed and confused

RIPE NCC DNSSEC Training Course material
http://www.ripe.net/training/dnssec/material/dnssec.pdf
I wish I had seen this before I started writing this article rather than after. If you are thinking of playing with DNSSEC this is well worth the time to read through – preferably before you start playing!.

Bind 9 Administrator Reference Manual, Internet Software Consortium, 2005
http://www.isc.org/index.pl?/sw/bind/arm93/
Chapter 4 of this guide describes DNSSEC. They've managed to do so in just 636 words, which is perhaps erring too far on the side of brevity. If you weren't familiar with DNSSC then this won't help much, and if you know all about DNSSEC tools already, then this will not contain anything you shouldn't already know!

DNS and Bind (5th Edition) , Paul Albitz &Cricket Liu, O'Reilly, 2006
Yes, this venerable bible of the DNS is now up to its 5th edition, which brings it into sync with Bind 9.3.2. Carefully written, loads of examples. It's a pity that I didn't have the 5th edition beside me when I was attempting this exercise, as Bind 9.3.2 redid its DNSSEC utilities, and the 4th edition of this book is, sadly, outdated in some critical details of DNSSEC utilities.

Pro DNS and BIND, Roy Aithison, Apress, 2005
I've heard really good things about this book, but I have yet to review a copy of this publication.

http://dnssec.nic.se
A description of DNSSEC and some details of the initiative to sign the .se top level domain. Actually this contains the .se zone key that I was looking for as a trust anchor last week!

http://www.dnssec.net
A resource guide for DNSSEC. A reasonably well ordered set of pointers to DNSSEC material.

Disclaimer

The views expressed are the author's and not those of APNIC, unless APNIC is specifically identified as the author of the communication. APNIC will not be legally responsible in contract, tort or otherwise for any statement made in this publication.

About the Author

Geoff Huston B.Sc., M.Sc., has been closely involved with the development of the Internet for many years, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. He is author of a number of Internet-related books, and has been active in the Internet Engineering Task Force for many years.

www.potaroo.net