six53

$ORIGIN a.DNSSEC.reference.card.

@ SOA six53.net. jpmens.six53.net. (20110329)

BIND authoritative

Configure BIND to serve DNSSEC data

```
options {
  dnssec-enable yes;
};
```

BIND Key generation

Create a KSK (flags == 257). Omit -f to create ZSK (256)

```
$ dnssec-keygen -a algo \
        -b bits \
        -f KSK \
        example.net
```

BIND smart signing -- needs cronjob

Keys are assumed in current directory. Option -o is origin; use -f for output file. Add to crontab(5)

```
$ dnssec-signzone -S \
      -o example.net zone.db
```

Add the resulting *zone.db.signed* to NSD or BIND

nsupdate

Update locally on server or using specific SIG(0) key

```
$ nsupdate -l
$ nsupdate -k Kmy.name*.private
```

Some commands

```
server addr [port]
zone zonename
prereq nxdomain domain
update delete name [ttl] [type [data]]
update add domain ttl type data
show
send
answer
```

Secure DDNS updates to BIND

Create a SIG(0) key pair

```
$ dnssec-keygen -C -a algo -b bits \
   -n HOST -T KEY my.name.
```

Add content of public .key to DNS

```
my.name. IN KEY 512 3 3 CLb...gaNM
```

Add an update policy to zone statement

```
update-policy {
  grant local-ddns zonesub ANY;
  grant my.name. zonesub A AAAA MX TXT;
};
```

BIND auto-sign -- fully automatic

Keys assumed in *mykeydir/* directory. Configure BIND to maintain signatures

```
zone "example.net" in {
  type master;
  key-directory "mykeydir";
  update-policy local;
  auto-dnssec maintain;
  sig-validity-interval 30; // days
  file "example.net";
};
```

Launch initial signing of zone

```
$ rndc signzone example.net
```

NSD authoritative

NSD requires no specific configuration to serve DNSSEC; it does so automatically for signed zones. Use BIND's signing tools

```
zone:
  name: "example.net"
  zonefile: "zone.db.signed"
```

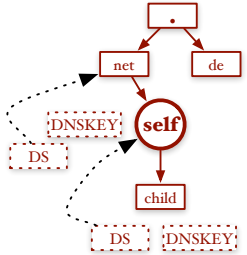
URLs

http://dnsviz.net/
http://dnscheck.iis.se/
http://dnssec-debugger.verisignlabs.com/
http://dnssec-or-not.net/
http://test-ipv6.com/
http://www.dnssec-failed.org/
http://www.dnssec-validator.cz/
http://www.zonecheck.fr/

BIND query log flags

```
+ S E T D C
```

- C — checking disabled
- D — DNSSEC requested
- T — TCP
- E — EDNS0 enabled
- S — request signed
- + — recursion requested



BIND validation

Configure BIND to perform DNSSEC validation

```
options {
  dnssec-enable yes;
  dnssec-validation auto;
  dnssec-lookaside auto; // DLV
};
```

Add KSK DNSKEYs if you have special validatable zones

```
trusted-keys {
  "my-domain" 257 3 7  "AWEAAQsF...";
};
```

Create static stub zone for your zone

```
zone "example.net" in {
  type static-stub;
  server-names { "localhost"; };
  server-addresses { 127.0.0.1; };
};
```

Unbound validation

Get and maintain root DNSSEC key

```
$ unbound-anchor -a root.key
```

Configure Unbound to perform DNSSEC validation

```
server:
  auto-trust-anchor-file: "root.key"
  dlv-anchor-file: "dlv.key"
```

If you want to add islands of trust, add DNSKEY and/or DS records in file, and add file to unbound.conf

```
  trust-anchor-file: "my.keys"
```

Optionally convince Unbound to query your non-delegated zone

```
stub-zone:
  name: "example.net"
  stub-host: localhost
  stub-addr: 127.0.0.1
```

dig

Useful dig options for DNSSEC queries.

```
$ dig @addr name type options

+dnssec        > sets DO requesting DNSSEC
+multiline              > verbosely human
+cd            > server shouldn't validate
+sigchase            > don't. Use drill(1)
```

Authenticated data: `;; flags: qr rd ra ad;`

drill

This is ldns' answer to dig(1)

```
$ drill [opts] name @addr type
```

Useful options

```
-D             > sets +DO requesting DNSSEC
-S                       > chase signatures
-k file      > DNSKEY or DS to verify sigs
```

Examples

```
$ drill -D example.net
$ drill -D -S -k root.key example.net
```

Acronyms & flags

DLV	DNS Lookaside Validation
DS	Delegation Signer
KSK	Key-Signing Key
ZSK	Zone-Signing Key
RFC	Request for Comments
do	flag: DNSSEC OK
ad	flag: authenticated data
qr	flag: query
aa	flag: authoritative answer
tc	flag: truncated
rd	flag: recursion desired
cd	flag: checking disabled
ra	flag: recursion available

RRs

Type	decimal
A	1
AAAA	28
AFSDB	18
APL	42
AXFR	252
CERT	37
CNAME	5
DHCID	49
DLV	32769
DNAME	39
DNSKEY	48
DS	43
HIP	55
IPSECKEY	45
IXFR	251
KX	36
LOC	29
MX	15
NAPTR	35
NS	2
NSEC	47
NSEC3	50
NSEC3PARAM	51
OPT	41
PTR	12
RRSIG	46
RP	17
SOA	6
SPF	99
SSHFP	44
TKEY	249
TSIG	250
TXT	16

Algorithm numbers

#	Mnemonic	RFC
3	DSA	3755
5	RSASHA1	3755
6	DSA-NSEC3-SHA1	5155
7	RSASHA1-NSEC3-SHA1	5155
8	RSASHA256	5702
10	RSASHA512	5702
12	ECC-GOST	5933

Digest algos

#	Desc
1	SHA-1
2	SHA-2
3	GOST

Credits

http://six53.net/refcard by @jpmens
IXFR from @miekg, @bortzmeyer