

Zone Key Tool

A signing and key admin tool for DNSSEC

Stiftelsen för Internetinfrastruktur

Stockholm
30. May 2008

Holger.Zuleger@hznet.de

Agenda

- Overview
 - DNS key generation with BIND
 - Basic ZKT features
 - Zone signing with BIND
 - ... and with the help of ZKT
- More detailed view on ZKT
 - Views
 - dynamic zones
 - Error logging
 - Key signing key rollover methods
 - RFC5011
 - KSK rollover
- Outlook
- References & Questions

DNSsec Overview

- BIND uses two commands for dnssec maintenance

- a. `dnssec-keygen` for key generation
- b. `dnssec-signzone` for zone signing

- `dnssec-keygen` requires some options to generate a key

```
$ dnssec-keygen -a RSASHA1 -b 1300 -n ZONE -f KSK example.net.
```

- ZKT uses a wrapper command plus a config file to simplify this a bit:

```
$ grep SK_ dnssec.conf
KSK_lifetime:    1y
KSK_algo:        RSASHA1 # (Algorithm ID 5)
KSK_bits:        1300
KSK_randfile:    "/dev/random"
ZSK_lifetime:    4w      # (2419200 seconds)
ZSK_algo:        RSASHA1 # (Algorithm ID 5)
ZSK_bits:        512
ZSK_randfile:    "/dev/urandom"
```

```
$ dnssec-zkt --ksk --create example.net.    ; generate ksk
```

```
$ dnssec-zkt -z -C example.net.           ; generate zsk
```

DNSsec Overview (keyfiles)

- A DNSSEC key is represented by two files (**public** and **private** part)

```
$ ls -l K*
-rw-r--r-- 1 hoz hoz 313 2008-05-07 00:57 Kexample.net.+005+27450.key
-rw----- 1 hoz hoz 1157 2008-05-06 17:43 Kexample.net.+005+27450.private
-rw-r--r-- 1 hoz hoz 177 2008-04-15 16:40 Kexample.net.+005+54680.key
-rw----- 1 hoz hoz 553 2008-04-15 18:40 Kexample.net.+005+54680.private
```

- The file entry has some infos about the key (**zone**, **tag**, **algorithm**, **date**)
But the type of key is coded in the flags field only
- `dnssec-zkt list` DNSKEYs in a more user friendly form

```
$ dnssec-zkt -a -t -f
Keyname          Tag Typ Sta Algorit          Age Lftm
example.net.    27450 KSK act RSASHA1    1w 2d23h34m55s<365d
example.net.    54680 ZSK act RSASHA1    4w 3d 7h52m17s!28d
```

- Some of the options are settable via the config file

```
$ sed -n '/zkt options/,/^$/p' /var/named/dnssec.conf | grep ":"
Zonedir:          "/var/named"
Recursive:        True
LeftJustify:      False
PrintTime:        True
PrintAge:         False
```

dnssec-zkt

- List all DNSKEYs found (a bit like `ls` for files)
Sorted by domain name, key type (KSK, ZSK) and date
- Is able to go recursive down a directory tree (Option `-r`)
- A directory or a key file could be specified as argument
Default directory is settable via `zonedir` parameter in `dnssec.conf`.
- Option `-p` print out the path name where the key files are found

```
$ dnssec-zkt -r -p -l example.net. .
Keyname                               Tag Typ Sta Algorit Generation Time
./views/intern/example.net./
    example.net. 00126 KSK act RSASHA1 Nov 20 2007 12:44:27
./views/extern/example.net./
    example.net. 23553 KSK act RSASHA1 Nov 20 2007 12:49:04
./views/intern/example.net./
    example.net. 05972 ZSK act RSASHA1 Nov 20 2007 12:44:27
./views/extern/example.net./
    example.net. 36122 ZSK act RSASHA1 Nov 20 2007 12:49:05
    example.net. 35744 ZSK pre RSASHA1 Dec 17 2007 23:45:27
```

dnssec-zkt (example output)

- Recursive key listing (sorted by domain name, key type and age)

```
$ dnssec-zkt -r -a examples
```

Keyname	Tag	Typ	Sta	Algorit	Generation	Time	Age
sub.example.de.	27321	KSK	act	RSASHA1	Apr 15 2008	18:40:55	5w 1d 6h36m23s
sub.example.de.	23742	ZSK	pre	RSAMD5	May 11 2008	23:32:01	1w 3d 1h45m17s
sub.example.de.	29194	ZSK	act	RSAMD5	May 09 2008	14:05:34	1w 3d 1h45m17s
example.de.	58635	KSK	rev	RSASHA1	Apr 23 2008	18:10:22	4w 9h 6m56s
example.de.	27450	KSK	act	RSASHA1	May 06 2008	17:43:29	2w 1d 19m56s
example.de.	17439	KSK	sta	RSASHA1	May 07 2008	00:57:22	2w 1d 19m56s
example.de.	54680	ZSK	act	RSASHA1	Apr 15 2008	18:40:55	5w 1d 8h37m18s
dyn.example.net.	09399	KSK	act	DSA	May 16 2008	12:39:19	5d12h37m59s
dyn.example.net.	46577	ZSK	act	RSASHA1	May 16 2008	12:39:19	5d12h37m59s
sub.example.net.	54876	KSK	act	RSASHA1	Oct 01 2007	08:24:24	33w 2d16h52m54s
sub.example.net.	01646	ZSK	act	RSAMD5	May 09 2008	13:59:11	1d13h47m16s
sub.example.net.	26431	ZSK	dep	RSAMD5	May 06 2008	15:25:28	1d13h47m16s
example.net.	41151	KSK	act	RSASHA1	Apr 20 2008	22:54:22	4w 3d 2h22m56s
example.net.	01764	KSK	sta	RSASHA1	May 06 2008	23:26:34	2w 1d 1h37m 3s
example.net.	05972	ZSK	act	RSASHA1	Nov 20 2007	12:44:27	26w 1d11h32m51s

- The state of a key is represented by the private key file name
 - e.g. state is pre-published (or standby) if private key file ends in `.published`
 - **First stage** of ZSK rollover: `sub.example.de`
 - **Last stage** of ZSK rollover: `sub.example.net`
 - **rfc5011 KSK** rollover in place: `example.de`

dnssec-zkt (Build in defaults)

- Print out the build in config options

```
$ dnssec-zkt --config "" -Z
#      @(#) dnssec.conf vT0.96 (c) Feb 2005 - May 2008 Holger Zuleger hznet.de

# dnssec-zkt options
Zonedir:      "."
Recursive:    False
PrintTime:    True
PrintAge:     False
LeftJustify:  False

# zone specific values
ResignInterval: 1w      # (604800 seconds)
Sigvalidity:   10d     # (864000 seconds)
Max_TTL:       8h      # (28800 seconds)
Propagation:   5m      # (300 seconds)
KEY_TTL:       4h      # (14400 seconds)
Serialformat:  incremental

# signing key parameters
KSK_lifetime:  1y      # (31536000 seconds)
KSK_algo:      RSASHA1 # (Algorithm ID 5)
KSK_bits:      1300
KSK_randfile:  "/dev/urandom"
ZSK_lifetime:  30d     # (2592000 seconds)
ZSK_algo:      RSASHA1 # (Algorithm ID 5)
ZSK_bits:      512
ZSK_randfile:  "/dev/urandom"

# dnssec-signer options
Keyfile:       "dnskey.db"
Zonefile:      "zone.db"
DLV_Domain:    ""
Sig_Pseudorand: True
```

Zone signing with BIND

- `dnssec-signzone` adds RRSIG (and NSEC) records to a zone
Output will be written to a `.signed` file
- DNSKEY RR should be added to the zone file via `$INCLUDE` directive
- By default, all key files in the current directory will be used for signing
Pay attention of the key signing key flag
- All signature records have a limited lifetime (default 30 days)
Settable via option `-e +172800` (from now-1h to now+48h)
- You have to do a resigning before the signature expire
- Until bind 9.4 there is no way to increment the serial number
This will prevent the usage for regular signing via a cron job
- Creates `dssset` and `keyset` files containing DS and DNSKEY RR
- A signed zone requires round about 8 to 20 files
It's recommended to use a separate directory for each zone

Zone signing with ZKT

- A wrapper command (`dnssec-signer`) is used for zone signing
 - Is able to increment the serial number
 - Supports plain integer, YYYYMMDDnn and unixtime (BIND9.4) format
 - Adds all the necessary DNSKEY RR to the zone file
 - Via `$INCLUDE` of generic `dnskey.db` file
 - Use of all the signing parameter options of the `dnssec.conf` file
 - Runs the signing process only if needed (a bit like `make`)
 - Edited zone file, refresh of RRSIG, new keys added, etc
 - Track the status of the key and start a key rollover if required
- Typical Zone file

```
$TTL      7200
;        The serial number is left justified in a field of at least 10 chars!!
@        IN SOA  ns1.example.net. hostmaster.example.net. (
                                244          ; Serial
                                43200       ; Refresh
                                1800        ; Retry
                                2W          ; Expire
                                7200 )     ; Minimum

$INCLUDE dnskey.db
```

dnssec-signer features

- Designed to use a separate directory for each zone
- Is very verbose (`-v -v`) or very silent
- Could be used to sign a single zone (in the current directory)

```
$ dnssec-signer -v -v -o example.net.
```
- or a directory tree of zone files

```
$ dnssec-signer -v -v -D /var/named/zones/de.
```
- or even read the zones to sign from a `named.conf` file

```
$ dnssec-signer -v -v -N /var/named/named.conf
```
- Sign only zones which are already signed
You have to create the `zone.db.unsigned` file manually for bootstrapping
- Option `-f` force a re-signing of the zone
- Use option `-r` to trigger a reload of the zone (via `rndc`)
The reload will be triggered only if it's necessary (new `.signed` file)

- Overview
 - DNS key generation with BIND
 - Basic ZKT features
 - Zone signing with BIND
 - ... and with the help of ZKT
- More detailed view on ZKT
 - Views
 - dynamic zones
 - Error logging
 - Key signing key rollover methods
 - RFC5011
 - KSK rollover
- Outlook
- References & Questions

BIND view support

- Added on user request (Dez 2007)
- Basically realised by a command line switch `--view viewname`
 - All config options will be read from `dnssec-viewname.conf`
 - Additionally, `dnssec-signer -N` will honor only zones inside a view config option
- Instead of using the `--view` switch you could link the command name

```
$ ln ~/bin/dnssec-zkt ~/bin/dnssec-zkt-viewname
$ ln ~/bin/dnssec-signer ~/bin/dnssec-signer-viewname
```
- Be aware of **not** using the same zone file for different views
- Use different zone files and include the common RR via a separate file

```
cat intern/example.net./zone.db
@ 7200 IN SOA .....
$INCLUDE dnskey.db
$INCLUDE /fullpath/common/example.net/zone.db
```
- If you change the common file please „touch“ the view specific zone file manually!

Dynamic Zone support (Experimental)

- Option `-d` added to `dnssec-signer` at 1st of April 2007
Signed zone files are named `.dsigned`
- BIND `named` signs new RR on demand with online zone signing key
And, for sure, adds NSEC records
- Regular re-signing is **not** performed by `named`
- We have to use the already signed zone as input file for zone signing
- ZKT copy signed file `zone.db.dsigned` to `zone.db` for re-signing
Replaces DNSKEY RR with new ones in case of key rollover
- Based on the BIND 9.4 feature to increment the soa serial number
`dnssec-signzone -N increment`
- Do a freeze/unfreeze on the zone while re-signing
Will be done by `dnssec-signer`
- With v0.96 the initial setup is simplified
Just create an empty `.dsigned` file

dnssec-signer error logging

- Added on request by IIS.se (v0.95)
- Actually not public available
(waiting for testing and feedback)
- Option `ErrorLog` specifies a log *file* or log *directory*
 - An existing file will be overwritten by `dnssec-signer`
Only error log of **last** `dnssec-signer` run
 - If a directory is specified, different files for logging will be used
 - File name looks like `zkt-YYYY-mm-ddThhmmssZ.log`
UTC timezone
- `ErrorLog` is also settable via config file (default is none)
- Exit code of `dnssec-signer` reflects number of errors:
 - 0: no error
 - 1-64: The number of errors occurred (or more than 63)
 - 127: Fatal error

RFC5011 KSK rollover (Experimental)

- Currently under test (v0.96)
- Create a so called „standby key“
A standby key is a pre published KSK not used for signing
\$ dnssec-zkt --ksk --create test.example.net.
- If the lifetime of the active KSK is over
 - a. A new standby key will be created
 - b. The old standby key will be activated
 - c. The old active key will be revoked
 - d. After 30 days, revoked key will be removed from zone apex
- Revoking a key means to set **bit 8** in the flags field

```
example.de. IN DNSKEY 385 3 5 BQEAAAABDAEYYP2lsGo...= ;
                /      \
                110000001
```

- Pay attention: Changing the flag field results in a new key tag(id)

```
$ dnssec-zkt --nohead --list-dnskey --ksk -l example.de.
example.de. IN DNSKEY 385 3 5 (
    BQEAAAABDAEYYP2lsGob0e77EYYDqsr ... wQ==
) ; key id = 58763 (original key id = 58635)
```

Double Signature (KSK) Rollover

- If lifetime of KSK is over
 1. Generate a new KSK; Use both key signing keys for key signing
Wait until new key is known by resolver (propagation time + old key TTL)
 2. Send new DS-set (or keyset) to the parent
Wait until the DS is propagated + TTL of the old DS-RR
 3. Remove the old key
- Step two is the critical one
 - How to send the DS-set to the parent? (In an authenticated way!)
 - How long does it take until the key is loaded on **all** authoritative name server?
- Currently no automatic ksk rollover available
Only warning message is written into error logfile if ksk is expired
- Use manual KSK rollover feature of dnssec-zkt

```
$ dnssec-zkt --ksk-roll-phase1 example.net.    $ dnssec-zkt --ksk-newkey example.net.
$ dnssec-zkt --ksk-roll-phase2 example.net.    $ dnssec-zkt --ksk-publish example.net.
$ dnssec-zkt --ksk-roll-phase3 example.net.    $ dnssec-zkt --ksk-delkey example.net.
```

Double Signature Rollover (outlook)

- Open issues
 - Where do we know the old DS TTL from?
Don't want to do a dig lookup because this requires online connection
 - And what about the propagation delay?
 - Could we simply wait „long enough“?
 - How could we send the new key to the parent (EPP extension ?)
- Next steps
 - Writing code for a full automatic ksk rollover if parent is on the same host (and uses the Zone Key Tool)
 - Currently a ksk rollover „sends“ DS automatically to the "parent"
In hierarchical mode
 - Discussing the open issues with typical parent zone maintainer (Registrars)
So that's one of the reason why I'm here today

- Overview
 - DNS key generation with BIND
 - Basic ZKT features
 - Zone signing with BIND
 - ... and with the help of ZKT
- More detailed view on ZKT
 - Views
 - dynamic zones
 - Error logging
 - Key signing key rollover methods
 - RFC5011
 - KSK rollover
- Outlook
- References & Questions

Outlook (v1.0?)

- `dnssec-zkt`
 - Split into two separate commands
 - `dnssec-ls` for key listing
 - `dnssec-key` for key management
key creation, status change, manual key rollover and so on
- `dnssec-signer`
 - Full support of `zsk`, `ksk` and RFC5011 rollover
 - Getting `ttl` values (`min`, `max`, `key_min`, `key_max`) out of zone file
 - Use of `ldns`?
- `dnssec-tkmon`
 - A RFC5011 trust anchor monitor daemon
 - Helper command for validating resolver (`bind`, `unbound`, ...)
 - Track status of trusted keys and update trusted key list
- Renaming so that all commands have a prefix of `zkt-` ?
`zkt-ls`, `zkt-key`, `zkt-signer`, `zkt-tkmon`

- Overview
 - DNS key generation with BIND
 - Basic ZKT features
 - Zone signing with BIND
 - ... and with the help of ZKT
- More detailed view on ZKT
 - Views
 - dynamic zones
 - Error logging
 - Key signing key rollover methods
 - RFC5011
 - KSK rollover
- Outlook
- **References & Questions**

References

Olaf Kolkman, NLnetlabs, Ripe NCC

„DNSSEC Howto Ver 1.8.2“ (http://www.nlnetlabs.nl/dnssec_howto/)

Internet Systems Consortium

BIND v9 Administrator Reference Manual

(<http://www.isc.org/sw/bind/arm94/>)

RFCs 4033 (DNS Security Introduction and Requirements)

4034 (Resource Records for the DNS Security Extensions)

4035 (Protocol Modifications for the DNS Security Extensions)

4641 (DNSSEC Operational Practices)

5011 (Automated Updates of DNS Security Trust Anchors)

DNSSEC software & tools

ldns (<http://www.nlnetlabs.nl/ldns/index.html>)

unbound (<http://unbound.net/>)

dnssec-tools (<http://www.dnssec-tools.org/>)

RIPE (http://www.ripe.net/disi/dnssec_maint_tool/)

ZKT (<http://www.hznet.de/zkt/>)

Questions ?

Questions ?

Thank you very much
for your attention!

CONTENTS

.....	1
Agenda	2
DNSsec Overview	3
DNSsec Overview (keyfiles)	4
dnssec-zkt	5
dnssec-zkt (example output)	6
dnssec-zkt (Build in defaults)	7
Zone signing with BIND	8
Zone signing with ZKT	9
dnssec-signer features	10
.....	11
BIND view support	12
Dynamic Zone support (Experimental)	13
dnssec-signer error logging	14
RFC5011 KSK rollover (Experimental)	15
Double Signature (KSK) Rollover	16
Double Signature Rollover (outlook)	17
.....	18
Outlook (v1.0?)	19
.....	20
References	21
.....	22