Testing Resolver Implementations of RFC 5011 for the Root KSK Roll

Martin Hoffmann
ICANN’s (Original) KSK Roll Schedule

<table>
<thead>
<tr>
<th>Phase C</th>
<th>Phase D</th>
<th>Phase E</th>
<th>Phase F</th>
</tr>
</thead>
<tbody>
<tr>
<td>First SKR 2017 Q2</td>
<td>Publication 2017 Q3</td>
<td>Rollover 2017 Q4</td>
<td>Revocation 2018 Q1</td>
</tr>
<tr>
<td>1 2 3 4 5 6 7 8 9</td>
<td>1 2 3 4 5 6 7 8 9</td>
<td>1 2 3 4 5 6 7 8 9</td>
<td>1 2 3 4 5 6 7 8 9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KSK 2010 publish + sign</th>
<th>KSK 2010 publish + sign</th>
<th>KSK 2010 publish</th>
<th>KSK 2010 revoke + sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSK 2017 publish</td>
<td>KSK 2017 publish + sign</td>
<td>KSK 2017 publish + sign</td>
<td>KSK 2017 publish + sign</td>
</tr>
</tbody>
</table>

- **2017-07-11**
- **2018-03-22**

based on the [2017 KSK Rollover Operational Implementation Plan](#)
deckard

Test harness for DNS software.
### The Domain Name System (Complete View)

<table>
<thead>
<tr>
<th>sec.sec.</th>
<th>insec.sec.</th>
<th>sec.insec.</th>
<th>sec.insec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS</td>
<td>ns</td>
<td>NS</td>
<td>ns</td>
</tr>
<tr>
<td>ns AAAA 2001:db8::4</td>
<td>A 192.0.2.4</td>
<td>rootns AAAA 2001:db8::1</td>
<td>A 192.0.2.1</td>
</tr>
<tr>
<td>test TXT text</td>
<td></td>
<td>test TXT text</td>
<td></td>
</tr>
</tbody>
</table>
Test Scenarios

- Happy path
- Un-publish before signing
- Roll-back after signing
- Revocation of old key
- Early re-introduction of old key
- Un-revoked old key
- Late re-introduction of old key
- Missing new key
- Non-writeable state directory
- Resolver restarts
- Resolver restarts with non-writable state directory
- Late installation with old key only
- Late installation with both keys
- Post-roll installation with old key only
- Post-roll installation with new key only
- Happy path with forwarding to a non-validating resolver
- Happy path while forwarding to a non-DNSSEC resolver
<table>
<thead>
<tr>
<th>Version Description</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>First production release</td>
<td>1.0.0</td>
<td>2008-05-20</td>
</tr>
<tr>
<td>RFC 5011 since</td>
<td>1.4.0</td>
<td>2009-11-26</td>
</tr>
<tr>
<td>Latest release</td>
<td>1.6.8</td>
<td>2018-01-19</td>
</tr>
<tr>
<td>Current Debian Stable</td>
<td>1.6.0</td>
<td>2016-12-15</td>
</tr>
</tbody>
</table>
Unbound

Findings

Late Installation

- Only trusts the new trust anchor after the 30 days’ add hold-down.
- Even if the new trust anchor is provided on installation.
- Fixed in 1.6.5 (2017-08-21).

Re-introduction of Old Key

- accepts the old key after remove and add hold-downs.
Unbound

Operational

RFC 5011 needs to be explicitly enabled

- trust-anchor-file v. auto-trust-anchor-file

Non-writeable state directory

- initially: logs error and carries on until the next restart
- 1.5.4 (2015-07-09): logs error and stops.
## Bind 9

### Versions

<table>
<thead>
<tr>
<th>Version Description</th>
<th>Version</th>
<th>Date</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>First production release</td>
<td>9.0.0</td>
<td>2000-09-16</td>
<td></td>
</tr>
<tr>
<td>First still usable release*</td>
<td>9.6.2</td>
<td>2010-03-01</td>
<td>* supporting RSASHA256</td>
</tr>
<tr>
<td>RFC 5011 since</td>
<td>9.7.0</td>
<td>2010-02-16</td>
<td></td>
</tr>
<tr>
<td>… working since*</td>
<td>9.7.1</td>
<td>2010-06-17</td>
<td>* according to our tests</td>
</tr>
<tr>
<td>Latest release</td>
<td>9.12.0</td>
<td>2018-01-23</td>
<td></td>
</tr>
<tr>
<td>Current Debian Stable</td>
<td>9.10.3</td>
<td>2015-09-16</td>
<td></td>
</tr>
</tbody>
</table>
Bind 9

Findings

Re-introduction of Old Key

- initially: accepts the old key even before the hold-downs have passed
- 9.10.2 (2015-02-25): never accepts revoked key again

Post-roll Installation with Old Key

- initially: resolver goes insecure instead of bogus
- 9.10.4 (2016-04-28): fixed
Bind 9

Operational

RFC 5011 needs to be explicitly enabled

• trusted-keys v. managed-keys

Non-writeable state directory

initially:

• repeatedly logs error and carries on

• when restarted after roll: goes insecure


• logs error once, carries on

• if kept running until after roll: bogus
# Knot Resolver

## Versions

<table>
<thead>
<tr>
<th>Description</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>First production release</td>
<td>1.0.0</td>
<td>2016-06-21</td>
</tr>
<tr>
<td>RFC 5011 since</td>
<td>1.0.0*</td>
<td></td>
</tr>
<tr>
<td>Latest release</td>
<td>2.1.0</td>
<td>2018-02-16</td>
</tr>
<tr>
<td>Current Debian Stable</td>
<td>1.2.0</td>
<td>2017-01-25</td>
</tr>
</tbody>
</table>

* in our tests first working in 1.2.0, faulty in 1.2.2, then working again in 1.2.5.
Knot Resolver

Findings

Re-introduction of old key

- accepts the old key after remove and add hold-downs

Late installation with old key only

- 1.2.0: accepts the new key during the add hold-down but not after
- 1.2.5: accepts the new key for one day

Revocation of old key

- 1.5.0: accepts the old key for one day after removal from DNSKEY record
Knot Resolver

Operational

Trust anchors are always updated via RFC 5011 🥂

Non-writeable state directory

• initially: stops at start with permission denied

• 1.5.1: same message but keeps running and goes bogus one day late

• 2.0.0: stops at start again
Knot Resolver

Operational

Resolver restarts

• 1.2.0: add hold-down restarts with every restart if trust anchor is kept but config directory recreated, otherwise bogus three days after key roll.

• 2.0.0: fixed
Thank you!

Martin Hoffmann
martin@opennetlabs.com