Root Algorithm Rollover and Lab Experiment in BII

Davey Song, Kevin Gong / BII
2019-3-2 / Tokyo / Yeti DNS Workshop
Facts about ECC

• ECC has much smaller keys and signatures with equivalent or better key strength
  • 2048-bit RSA —> 251 bytes RRSIG RDATA, 260 bytes DNSKEY RDATA
  • ECDSA P-256 —> 59 bytes RRSIG RDATA, 64 bytes DNSKEY RDATA
  • ECC with 256-bit group ≈ RSA 3072-b

• ECDSA P-256 and P-384 are standardized for use in DNSSEC in RFC 6605 (2012)
Uncertainties of ECC for DNSSEC

• Impact of switching to ECC on resolvers is uncertain
  – Performance issue, up to an order of magnitude slower than RSA
  – ECC readiness of Resolvers (Large install base) is uncertain especially the Auto-Rollover from RSA to ECC
  – The right timing is vital for success of Algorithm rollover

"the new ZSK wasn’t pre-published long enough", "this resulted in validation errors" from Unbound-users mailing list
DNSSEC Algorithm Rollover approach

• Specified in RFC6781 and RFC4035, using double-signature rollover, expect one signature for each algorithm in the zone apex

The conservative approach interprets this section very strictly, meaning that it expects that every RRset has a valid signature for every algorithm signaled by the zone apex DNSKEY RRset, including RRsets in caches. The liberal approach uses a more loose interpretation of the section and limits the rule to RRsets in the zone at the authoritative name servers.

----section-4.1.4 of RFC6781

• Although RFC6781 recommend conservative approach, many open source signers like BIND "managed keys" and OpenDNSSEC implements the “liberal” approach.
DNSSEC Algorithm Rollover

• Experience provided by practice on level of second domain by RIPE NCC and TLD .BR, .SE,
  – RIPE NCC suggest to roll both ZSK and KSK (2015)
  – .SE Algo Roll adopted liberal approach with 6 failure out of 10,000 probes (2018)
• There is no existing experience on the level of Root (automatic algorithm rollover for trust anchors, RFC5011 considered)
• It is still interesting and unknown whether ZSK and KSK should be rolled at the same time
Algorithm rollover in Lab Environment

• To test potential configurations as many as possible
  – Both Conservative and liberal approaches
  – Roll KSK without ZSK, and Roll them at the same time

• Four test configurations are proposed
  – Test1: Republish KSK without signature as we rolled the key (Yeti KSK rollover), intentional violation of RFC6781
  – Test2: Similar with Test1 but republish KSK and its signature without rolling ZSK
  – Test3: Roll both ZSK and KSK in liberal approach
  – Test4: Roll both ZSk and KSK in conservative approach
Test Setup

• For each test, setup 3 authoritative servers
  – 1 Master : BIND 9.11.5-P1
  – 2 Slave: Knot 2.7.6, NSD 4.1.26
  – Set DNSKEY TTL: 600 seconds

• For each test, setup 2 resolvers
  – BIND 9.11.5-P1, Unbound 1.8.3

• Monitoring setup
  – Check rfc5011 state by recording the managed.key file on two resolver (managed.key file)
  – Monitor the trust chain by recording the response for random/junk queries to see whether the AD bit is set for a valid response
  – Monitoring the changes of Root zone (DNSKEY record and signature)
  – Capture DNS packet via dnscap on all servers
Fast Algorithm rollover in 10 minutes

• Since RFC5011 timer ( wait 30 days to trust a new KSK) is too long, we hack the resolver to accept a shorter timer to get a result in a stand-on time
  – Add Hold-Down Time: 60 second
  – Remove Hold-Down Time : 60 second

Restart Bind9:
# named -c /etc/named.conf -t /var/named -u named -T mkeytimers=2/5/60

Edit unbound.conf:
add-holddown:60
del-holddown:60
permit-small-holddown:yes
keep-missing:300

All Tests got passed on fast algorithm rollover!
Test1: Timeline and results

- Original design: 10 days for each slot
- We just wait 30 days to and manually check if the key is trusted in resolver’s “managed.key” file and the validation status
- An accidental mistake ZSK become inactive in slot 5 before the new key trusted. It resulted validation failure
- During slot 5 SERVFAIL for BIND resolver and No AD bit set in Unbound resolver (with ‘val-permissive-mode: yes’)
- RFC5011 ...OK
Test2: Timeline and result

- Pass the test!
- Both BIND and unbound accept and trust the new key and new algorithm when 30-day timer expires
- The validation tests got passed during the whole process (slot 6, slot 7 and slot 8)
### Test3: Timeline and result

<table>
<thead>
<tr>
<th>Slot 1</th>
<th>Slot 2</th>
<th>Slot 3</th>
<th>Slot 4</th>
<th>Slot 5</th>
<th>Slot 6</th>
<th>Slot 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>old KSK</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>Revoke+sign</td>
<td></td>
</tr>
<tr>
<td>new KSK</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
</tr>
<tr>
<td>old ZSK</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
</tr>
<tr>
<td>New ZSK</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
</tr>
</tbody>
</table>

- **An accidental mistake:** KSK become inactive in slot 5 before the new key trusted. It resulted validation failure for both BIND and Unbound resolver.
- During slot5 SERVFAIL for BIND resolver and No AD bit set in Unbound resolver (with ‘val-permissive-mode: yes’)
- BIND restart the Add Hold-Down Time for another 30 days
- Unbound continue the timer and trusted the new key after the timer expired
### Test4: Timeline and result

<table>
<thead>
<tr>
<th></th>
<th>slot 1</th>
<th>slot 2</th>
<th>slot 3</th>
<th>slot 4</th>
<th>slot 5</th>
<th>slot 6</th>
<th>slot 7</th>
<th>slot 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>old KSK</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>Revoke+sign</td>
<td></td>
<td></td>
</tr>
<tr>
<td>new KSK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pub+sign</td>
<td>pub+sign</td>
</tr>
<tr>
<td>old ZSK</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td>pub+sign</td>
<td></td>
<td>sign</td>
</tr>
<tr>
<td>New ZSK</td>
<td>sign</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>pub+sign</td>
<td>pub+sign</td>
</tr>
</tbody>
</table>

- Test4 results the same with Test3 as well as the **accidental mistake**
Conclusion

• All Tests supposed to be passed if there is no key timing error even for test 1
• Future tests should be done
  – Test roll back if failure observed
  – Test stand-by Key
• Invite more resolvers to join