

Root Algorithm Rollover and Lab Experiment update

Davey Song, Kevin Gong

2019-7-21/ Montreal / Yeti DNS Workshop

ECDSA-P256 vs RSASHA256-2048

Type	ECDSAP256SHA256 (13)	RSASHA256 (8)
DNSKEY-RR-size (byte)	79	275
RRSIG-DNSKEY-RR-size (byte)	94	286
RRSIG-Signature-size (byte)	59	251

rfc8624:

ECDSAP256SHA256 provides more cryptographic strength with a shorter signature length than either RSASHA256 or RSASHA512. ECDSAP256SHA256 has been widely deployed; therefore, it is now at MUST level for both validation and signing. It is RECOMMENDED to use the deterministic digital signature generation procedure of the Elliptic Curve Digital Signature Algorithm (ECDSA), specified in [[RFC6979](#)], when implementing ECDSAP256SHA256 (and ECDSAP384SHA384).

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

- OS: Ubuntu 16.04.5 LTS
- 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 3 resolvers
 - BIND 9.11.5-P1
 - Unbound 1.8.3
 - pdns-recursor 4.2.0~alpha1(manual configure KSK), pdns-recursor 4.0.0~alpha2
- DNSKEY
 - Add a standby-KSK
- Monitoring setup
 - Check rfc5011 state by recording the managed.key file on two resolvers (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

Test1: Timeline and results

	slot 1	slot 2	slot 3	slot 4	slot 5	slot 6	slot 7
old KSK	pub+sign	pub+sign	pub+sign	pub+sign	pub	revoke+sign	
new KSK		pub	pub	pub	pub+sign	pub+sign	pub+sign
stand-by KSK		pub	pub	pub	pub	pub	pub

- Note: slot 1,2,3,4 and 7 are 10 days. slot 5 and slot 6 are 1 day
- **PASS**
- We just wait 30 days to and manually check if the key is trusted in resolver's "*managed.key*" file and the validation status
- RFC5011 ...OK

Test2: Timeline and result

	slot 1	slot 2	slot 3	slot 4	slot 5	slot 6	slot 7
old KSK	pub+sign	pub+sign	pub+sign	pub+sign	pub	Revoke+sign	
new KSK		pub+sign	pub+sign	pub+sign	pub+sign	pub+sign	pub+sign
stand-by KSK		pub	pub	pub	pub	pub	pub

- Note: slot 1,2,3,4 and 7 are 10 days. slot 5 and slot 6 are 1 day
- **PASS**
- 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 (slot6 , slot 7)

Test3: Timeline and result

	slot 1	slot 2	slot 3	slot 4	slot 5	slot 6
old KSK	pub+sign	pub+sign	pub+sign	pub+sign	Revoke+sign	
new KSK		pub+sign	pub+sign	pub+sign	pub+sign	pub+sign
stand-by KSK		pub	pub	pub	pub	pub
old ZSK	pub+sign	pub+sign	pub+sign	pub+sign	pub+sign	
New ZSK		pub+sign	pub+sign	pub+sign	pub+sign	pub+sign

- Note: slot 1,2,3,4 and 6 are 10 days. slot 5 is 1 day.

- PASS
- Repeat the servail
 - 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

	slot 1	slot 2	slot 3	slot 4	slot 5	slot 6	slot 7	slot 8
old KSK	pub+sign	pub+sign	pub+sign	pub+sign	pub+sign	Revoke+sign		
new KSK			pub+sign	pub+sign	pub+sign	pub+sign	pub+sign	pub+sign
stand-by KSK			pub	pub	pub	pub	pub	pub
old ZSK	pub+sign	pub+sign	pub+sign	pub+sign	pub+sign	pub+sign	sign	
New ZSK		sign	pub+sign	pub+sign	pub+sign	pub+sign	pub+sign	pub+sign

- Note: slot 1,3,4,5 and 8 are 10 days and slot 2, 6 and 7 are 1 day. It means new/old zsk sign action and the old KSK revoke+sign action only last 1 day.

- PASS

Conclusion (1)

1. 4 different algo-roll schemes proceeded and rolled smoothly with 3 typical resolver.
2. Difference behaviors between BIND and Unbound NO.1: KSK-inactive
3. Algorithm rollover with only KSK works for the resolver in Case 1 (with -P option) and Case 2. The BIND and Unbound can work with it without change on ZSK.
4. Double-DS rolling scheme(prepublish the new algorithm key without signing with it) is testing against the protocol. It was observed that zone can be signed using dnssec-signzone with -P option, and the resolvers accept this approach.
5. Stand-by Key works fine and can be switched immediately if "incumbent" key is not available in case of failure
6. PowerDNS (pdns-recursor 4.0.0~alpha2-2ubuntu0.1) resolver failed when configuring multiple algorithm KSK

Conclusion (2)

- DNSKEY packet size stats

	slot1	slot3	Slot6(Revoke)
RSASHA256-only	864	1975	1975
Case3	864	1195	1481(ZSK signed the DNSKEY RR)
ECDSA-only	280	611	611

Slot1: RSASHA256-2048*2, RRSIG-RSA*1

Slot3: ECDSAP256SHA256*3, RSASHA256-2048*2, RRSIG-RSA*1, RRSIG-ECDSA*1

Slot6: ECDSAP256SHA256*3, RSASHA256-2048*2, RRSIG-RSA*2, RRSIG-ECDSA*1

dnssec-signzone:

when old RSA KSK is revoked, the RSA ZSK will be used to sign both the zone and DNSKEY records, even when -x option is set. -x option means “only sign the DNSKEY RRset with key-signing keys”.

Future

Run it on Yeti testbed?