Signaling between Signer and Enforcer

Yuri, Matthijs

November 28, 2011

Since the Signer is smart about rollovers (i.e. provide gradual transition between two keys) and the Enforcer supports ludicrous rollovers, we must take extra care in how the two communicate.

The Enforcer has the following directives in the signconf:

Key Section The key is mentioned in the configuration. Signatures with this key may be reused.

Publish Flag If set, a key must have its DNSKEY record published.

KSK Flag If set, DNSKEY set must be signed using this key.

ZSK Flag If set, records must be signed using this key

1 ZSK Flag

This flag is problematic. In order to spread load, the signer keeps using signatures of old keys which are not yet expired before generating signatures with the new key. The Enforcer does not necessary performs a roll between 2 keys but rather N keys. Additionally a zone might be signed with multiple key of the same algorithm.

The challenge is for the signer to find out when new signatures must be generated and with which keys. Figure 1 shows the decision tree Matthijs and I figured out.

The signer first of all loops through all RR sets, then over all keys. Each iteration the tree is consulted. We can define 4 different checks:

Active? The key has ZSK Flag.

Signature? This RR set already has a signature for this key.

Valid? Said signature is still valid.

Inactive key with valid Signature? There is a still valid signature, for an old key.

2 Behavior

This specification leads to the following behavior:

- Smooth rolling of ZSKs is still supported.

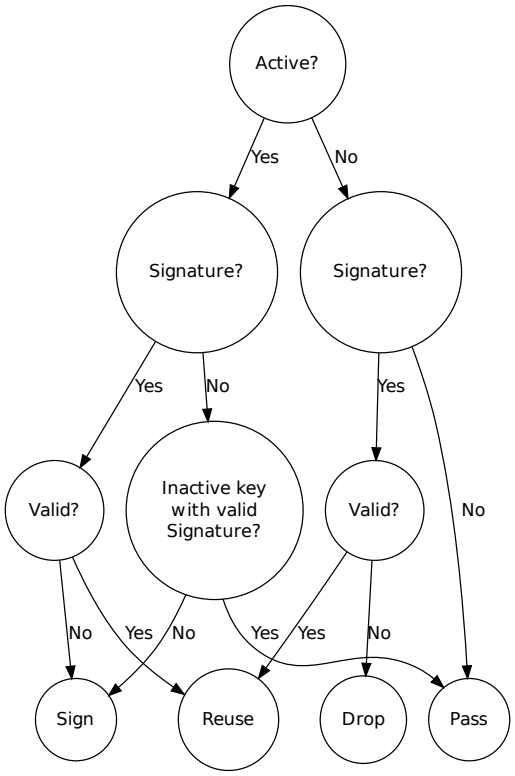

Figure 1: What to do with this RRset for this key?

- When rolling from 1 ZSK to N, Any of those N will be used only after the old signatures expire.
- When signing with key A and then also signal to use key B. B's signatures will be introduced immediately.
- When signing with N keys, we can replace one of those in a smooth way.