DNSOP Working Group
INTERNET-DRAFT

Olaf M. Kolkman
RIPE NCC
Miek Gieben
NLnet Labs
Roy Arends
Nominum

June 2001

Rollover of statically configured resolver keys.
<draft-ietf-dnsop-resolver-rollover-00.txt>

Status of this Memo

Comments should be sent to the authors or to the dnsop WG mailing list dnsop@cafax.se.

Copyright Notice

INTERNET-DRAFT draft-ietf-dnsop-resolver-rollover-00.txt June 2001

Abstract

Key rollovers will be needed for secure deployment of the DNS security extensions (DNSSEC). From an end-user perspective these rollovers should be transparent i.e. at any point in time an end-user should be able to verify the chain of trust from a statically configured secure entry point.

When a zone is being used as the secure entry point for one or more end-users then a rollover of the keys from that zone will need to result in a reconfiguration of the keys at the end-user resolvers.

We propose a simple polling mechanism that can be used for auto-reconfiguration of statically configured keys in end-user resolvers.

Table of content

11.. SSccooppee aanndd RRaattiioonnaallee

"The Domain Name Security Extensions" (DNSSEC) [RFC2535] is a means to provide data integrity and authentication to the DNS. Using signatures over DK RRsets [IDdkey] end users can build chains of trust from from statically configured roots of secure island [RFC3090] to the data in the DNS that needs to be verified. In the remainder of this document we will refer to 'statically configured roots of secure islands' as secure entry points.

[Author's note: This draft assumes the DK record is used for delegating authority from parent to child but does not rely this particular way of parent-child authority delegation]

Key rollover is the process where a ZONE key [RFC3090 section 2,2a] is replaced by another ZONE key. Since Public/Private keys have a limited life time, key rollovers need to happen at regular intervals [RFC2541]. These rollovers are normally referred to as scheduled key rollovers. Emergency key rollovers, where a key needs to be replaced by another key because a private key has been compromised, are not the subject of this document.

During DNSSEC operations an end-user(*) follows a chain of trust from one of their statically configured security entry points to the data that needs to be verified. The secure entry point keys are obtained by an initial key exchange. Initial key exchanges are outside the scope of this document. For the end-users it is important that existing chains of trust from the secure entry point to the data somewhere in the DNS remains intact when a zone in the the chain of trust perform a key rollover.

The rollover of keys for zones that are configured as secure entry points may happen frequently. As long as the root is not secure, multiple TLD and GTLDs will act as secure entry points, the 'default' zone will in general also be configured as secure entry point so that one does not rely on connectivity to it's parent.

Zone administrators will not have a-priory knowledge about which

---

* In the context of this document an end-user is an entity that does the verification of the chain of trust. This could be a stub resolver but also a caching forwarder.

resolvers have their zones configured as secure entry points so it will be impossible for a zone administrator to contact all end-user resolvers when a key exchange is to commence commence.

22.. GGeenneerraall DDeessccrriippttiioonn ooff KKEEYY RRoolllloovveerr..

From an end-user's point of view there are two types of rollover.

Parent Child rollovers (PC-rollovers), where the zone that rolls over is part of a chain of trust and has authority delegated from a parent zone, and

Secure Entry rollovers (SE-rollovers) where the end-user resolver is configured with the key of the zone that rolls over itself. In other words the zone is a secure entry point for the end-user.

For zone administrators it is clear they are involved in a PC-rollover; They will have to get get their parent to create a new DK record. For some zones it may not be obvious that their KEYs are configured statically at end-user hosts, they will need to enable a SE-rollover.

Key rollovers of the root will always be of SE-rollovers. Key

rollovers of GTLDs and TLDs are likely to be SE-rollovers.

The requirements and policies for a PC-rollovers are somewhat different from those of a SE-rollovers. In both cases the zone administrator decides to rollover it's key and in both cases another party has to take specific action.

During a PC-rollover the old an the new key have to coexist in the zone and the zone must be signed with both the old and new key so that end-users can follow the chain of trust from a secure entry point parent downward. This is needed because the DK record change needs some time to propagate through the DNS and during this time there will be DK records in the DNS that point to the old key and DK records that point to the new key. Once the parent's new DK record has been distributed to all the authoritative servers and one is sure the old parent data should have timed out from caches the old key can be removed from the zone.

If authority has not been delegated from a parent i.e. the zone

administrator is sure that the rollover is only relevant to resolvers. then the old key may immediately be replaced by a new KEY, the key set containing the new key MUST then be signed with both the old and new key so that resolvers can verify the new KEY against the statically configured old KEY.

33.. PPeerriiooddiicc PPooll lliinngg bbyy rreessoollvveerrss..

To notice an ongoing key rollover the resolver will need to periodically query for the KEY RRset for the zone it has configured as a secure entry point, if the ZONE KEYS published in the apex of the zone have changed with respect to the statically configured keys a rollover is ongoing.

To detect new Zone KEY in the apex of a zone, the resolver uses the same mechanism as slave nameservers use for detecting changes to a primary zone (section 4.3.5 of [RFC1035]); The resolver should check the SOA RR by polling the authoritative server periodically. As

soon as the SOA serial has been increased, a query for the KEY RRset must be made. The resolver then proceeds by verifying the KEY RRset against one of the existing statically configured keys. The KEY RRset is also verified against the self signatures made with the ZONE keys from the KEY RRset.

If both the verifications are successful, the ZONE Keys from the KEY RRset are compared against the existing statically configured keys, if these two sets of keys differ a rollover is taking place and the statically configured KEYs are to be replaced by the ZONE keys from the d KEY RRset. The application MAY send a notification to the resolver administrator who might want to audit the rollover using an off-band mechanism. If one of the verifications fail the application SHOULD send a warning to the resolver administrator.

If the resolver finds it impossible to perform the serial check for the EXPIRE interval it MUST not discard existing statically configured keys, the application SHOULD send a warning to the resolver administrator who might wish to unconfigure the key.

44.. ZZoonnee aaddmmiinniissttrraattiioonn ccoonnssiiddeerraattiioonnss..

If zones are configured as secure entry points then a SE-rollover

takes place. Zone administrators have to take the following into account for successful auto-configuration of end-users resolvers.

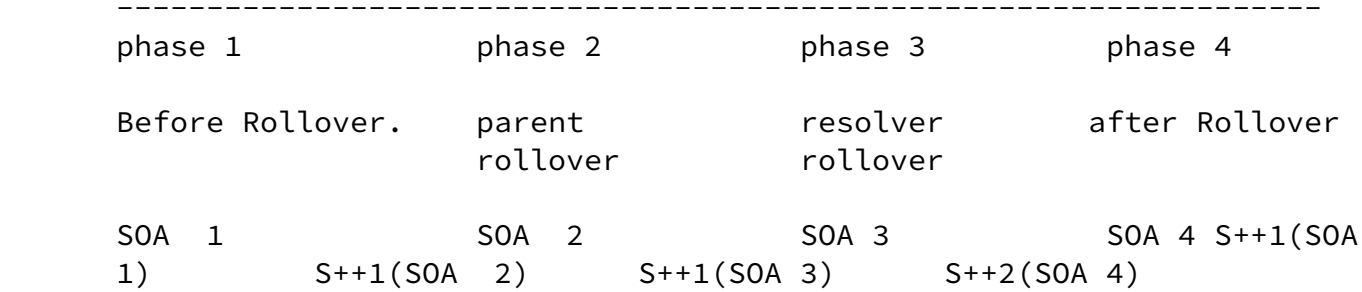
Since resolvers may not be able to poll the KEY RRset for extended times, the period resolvers have access to the new key should be made as long as possible.

The time during which the parent zone changes the delegation of authority from the old key to the new key can be relatively short (phase 2 in figure 1). The length of this time interval is determined by the TTL of the parents signature and the REFRESH interval in the parents SOA; all authoritative slave servers must have had the change to load the new DK record and the old DK record must have expired from caches. During this interval the child zone needs to publish two KEYs and signatures made with both the keys.

A zone administrator may decide to sign their zone with the old key for an extended period of time (phase 3). During this time resolvers that use the zone as a secure entry point be able to verify the zone with the old key and will still be able to grab the new key. Resolvers that do not have the zone configured as a secure entry point will use the new key when walking the secure tree from the secure entry point via the parent to the zone.

Since the intention of a key rollover is to stop using the key there will be a phase 4 where the zone is signed with the new key only. If resolvers that have the zone configured as a secure entry point have not changed the statically configured key the zone and the it's sub zones will become "BAD".

Note that during a SE-rollover, i.e. a rollover for zones that are not part of a chain of trust, phase 2 may be skipped. The root would be an example of such a zone. For these zones a resolver that can dynamically update itself, should always have the same statically configured ZONE KEYs as the ZONE KEYs published in the zone itself.



```
                    S++2(SOA  2)      S++2(SOA 3)

K++1                K++1              K++2              K++2
S++1(K++1)          K++2              S++1(K++2)        S++2(K++2)
                    S++1(K++1,K++2)   S++2(K++2)
                    S++2(K++1,K++2)
```

Figure 1:Zone status during rollover

Key and signature notation is explained in the appendix. The number behind the SOA indicates it's serial number.

55.. SSeeccuurriittyy CCoonnssiiddeerraattiioonnss

The key rollover described are based on the verification of new key material against an existing chain of trust. This existing chain of trust can be broken; this could be the case when there was an unnoticed attack during the initial key exchange or when one of the keys against which is verified has been compromised. Resolver administrators should regularly audit statically configured keys against their origin using data that is not published via the DNS.

If a key that is statically configured has been compromised, a rollover of statically configured keys of a resolver may be performed by the attacker. To be successful the attacker has to spoof the nameserver or pollute a caching forwarder that the end-user uses to obtain the new keys. To limit damage of a compromised key, the mechanism described here MAY still be used to distribute a new key during an emergency key rollover. Resolvers that are able to get the key from the original nameserver or from an unpolluted cache will not be

vulnerable to an attack with compromised keys after the rollover.

66.. RReeffeerreennccees

[IDdkey] "Delegation Signer Record in Parent", draft-ietf-dnsext-delegation-signer-00.txt, O. Gudmundsson May 30 2001.

[RFC1035] "Domain Names - Implementation and Specification", P. Mockapetris. November 1987.

[RFC2535] "Domain Name System Security Extensions", D. Eastlake. March 1999.

[RFC2541] "DNS Security Operational Considerations", D. Eastlake. March 1999.

[RFC3090] "DNS Security Extensions Clarification on Zone Status", E. Lewis. March 2001.

77.. AAuutthhoorrss'' AAddddrreessssseess::

Olaf M. Kolkman	Miek Gieben	Roy Arends
RIPE NCC	Stichting NLnet Labs	Nominum
OKolkman@ripe.net	Miek@nlnetlabs.nl	Roy.Arends@nominum.com
http://www.ripe.net	http://www.nlnetlabs.nl	http://www.nominum.com

88.. AAppppeennddiixx:: NNoottaattiioonn

In this draft we use the following notation:

A Key is identified by K<owner>+<protocol>+<keytag>. The owner, protocol and keytag are optional if their value is clear from the context or when their value is of no importance. So Kfoo.example+3+1 is the DSA Key with keytag 1 belonging to label foo.example. K++1 and K++2 are two keys from the same zone and algorithm, both not relevant or clear from the context, with different keytags.

A Signature is identified as S<ownername>+<protocol>+<keytag>(<RR identifier>). So Sfoo.example+2+1(www.foo.example A) is the signature made with the foo.example algorithm 2 (DSA) key with keytag 1, over the www.foo.example A record.