Domain Name System Operations
Internet-Draft
Intended status: Standards Track
Expires: February 22, 2010

W. Wijngaards
O. Kolkman
NLnet Labs

August 21, 2009

DNSSEC Trust Anchor History Service
draft-wijngaards-dnsop-trust-history-02

Status of This Memo

Copyright Notice

Abstract

When DNS validators have trusted keys, but have been offline for a longer period, key rollover will fail and they are stuck with stale trust anchors. History service allows validators to query for older

DNSKEY RRsets and pick up the rollover trail where they left off.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1. Introduction

This memo defines a trust history service for DNS validators -- the component in a resolver that performs DNSSEC [RFC4034] validation, validator for short.

A validator that has been offline or missed an (emergency) rollover can use this service to reconfigure themselves with the current trust-anchor. Using a newly defined resource record (RR) that links old DNSKEYS together, the TALINK RR, a validator fetches old DNSKEY RRsets and checks they form a chain to the latest key (see Section 2). The lists of old DNSKEYS, linked with the TALINK RRs, do not necessarily need to be published in the zone for which the DNSKEY history is being maintained but can be published in any DNS domain. We will call the entity that offers the trust history the History Provider. The choice of the History Provider is made by the maintainer of the validator, possibly based on a hint provided, using the TALINK, by the zone owner.

The algorithm that the validator uses to construct a history and reconfigure a new key is detailed in Section 3. The algorithms for how providers of trust history can fetch the DNSKEY data as published by the zone they track, and publish those in their own domain, are explained in Section 4.

2. The TALINK Resource Record

The DNS Resource Record type TALINK (decimal TBD) ties the elements of a linked list of DNSKEY RRs together.

The rdata consists of two domain names. The first name is the start, or previous name, and the other name the end or next name in the list. The empty label '.' is used at the endpoints of the list.

The presentation format is the two domain names. The wire encoding is the two domain names, with no compression so the type can be treated according to [RFC3597]. The list is a double linked list, because this empowers low memory hosts to perform consistency checks.

3. Trust History Lookup

This is the algorithm that a validator uses to detect and resolve the situation in which a trust-anchor is out of sync with the DNSKEYs published by a zone owner. The algorithm uses the TALINK RR type which is used to link various old DNSKEYs as published by the History Provider, to arive from the outdated configured Trust Anchor to one that matches the current DNSKEY. The TALINK RR type is defined in Section 2.

When the algorithm below results in failure the trust history cannot be build and a new trust anchor will need to be re-configured using another mechanism.

Step 1: The validator performs a DNSKEY lookup to the target zone, which looks like any other initial DNSKEY lookup that the validator needs to match a trust anchor to the currently used DNSKEY RR set. If the keyset verifies with the trust anchor currently held, the trust-anchor is not out-of sync. Otherwise, store the DNSKEY RR set as result. The algorithm will succesfully build a linked list to this DNSKEY RR or fail.

All nameservers (the ones authoritative for the zone or the upstream resolver caches when the validator is not full resolver) SHOULD be checked to make sure the DNSKEY RR sets are the same. The results can differ if a key-rollover is in progress and not all nameservers are in sync yet. This case can be detected by checking that the older keyset signs the newer and take the newer as result keyset. The keysets are distinguished by the average over the middle of the inception and expiration dates of the signatures that are validated by the keyset itself. Otherwise, the lookup fails. If the check fails then the inconsistency may be the result of spoofing, or compromise of (DNS) infrastructure elements.

Step 2: Fetch the trust history list end points. Perform a query of QTYPE=TALINK and QNAME being the domain name that is configured to be the History Provider for the particular domain you are trying to update the trust-anchor for.

Step 3: Go backwards through the trust history list as provided by the TALINK linked list. Verify that the keyset validly signs the next keyset. This is [RFC4034] validation, but the RRSIG expiration date is ignored. [Editor note: Are we shure that there are no server implementations that will not serve expired RRSIG, are such 'smart' servers allowed by the specs? In other words do we need clarification in the DNSSEC-updates document?] Replace the owner domain name with the target zone name for verification.

One of the keys that signs the next keyset MUST have the SEP bit set. The middle of inception and expiration date from the valid signature MUST be older than the signature checked last time. Query type TALINK to get previous and next locations.

If all SEP keys have the REVOKE flag set at this step, and the keyset is signed by all SEP keys, then continue but store that the end result is that the trust point is deleted, see Section 5 [RFC5011].

If all SEP keys are of an unknown algorithm at this step, continue and at the next step, when you verify if the keyset signs validly: if false, continue with result a failure, if true, continue with the end result that the trust point is deleted. Thus, the keysets with unknown algorithms are stepped over with an end result of failure because the validator cannot determine if unknown algorithm signatures are valid, until the oldest keyset with unknown algorithms is signed by a known algorithm and the result is set to deletion and step 3 continues to a known key.

Step 4: When the trust anchor currently held by the validator verifies the keyset, the algorithm is done. The validator SHOULD store the result on stable storage. Use the new trust anchor for validation (if using [RFC5011], put it in state VALID).

4. Trust History Tracker

External trackers can poll the target zone DNSKEY RRset regularly. Ignore date changes in the RRSIG. Ignore changes to keys with no SEP flag. Copy the newly polled DNSKEY RRset and RRSIGs, change the owner name to a new name at the history location. Publish the new RRset and TALINK records to make it the last element in the list. Update the TALINK in the target zone that advertises the first and last name.

Integrated into the rollover, the keys are stored in the history and the TALINK is updated when a new key is used in the rollover process. This gives the TALINK and new historical key time to propagate.

The signer can support trust history. Trust history key sets need only contain SEP keys. To use older signers, move historical RRSIGs to another file. Sign the zone, including the TALINK and DNSKEY records. Append the historical RRSIGs to the result. Signing the zone like this obviates the need for changes to signer and server software.

5. Example

In this example example.com is the History Provider for example.net. The DNSKEY rdata and RRSIG rdata is omitted for brevity, it is a copy and paste of the data from example.net.

```
$ORIGIN example.com.
example.com. TALINK h0.example.com. h2.example.com.

h0 TALINK . h1.example.com.
h0 DNSKEY ...
h0 RRSIG ...

h1 TALINK h0.example.com. h2.example.com.
h1 DNSKEY ...
h1 RRSIG ...

h2 TALINK h1.example.com. .
h2 DNSKEY ...
h2 RRSIG ...
```

The example.net zone can advertise the example.com History Provider by providing the TALINK shown here at example.com at the apex of the example.net zone. The TALINK at example.com is then not needed.

6. Deployment

The trust history is advertised with TALINK RRs at the zone apex. These represent alternative history sources, that can be searched in turn. The TALINK at the zone apex contains the first and last name of the list of historical keys.

The History Provider decides the oldest age keys it wants to publish, as operations permit. If validators no longer have trust in the keys then they need no longer be published. The oldest key entries can be omitted from the list to shorten it.

The validator decides how long it trusts a key. A recommendation from the zone owner can be configured for keys of that zone, or recommendations per algorithm and key size can be used (e.g. see [NIST800-57]). If a key is older than that, trust history lookup fails with it.

The history lookup can be used on its own. Then, the trust history is used whenever the key rolls over and no polling is performed. The time of the last successful key lookup is stored on stable storage. The trust history algorithm is not started unless the last successful key lookup was more than x time ago. This time is suggested to be

the same has the Hold-Down timer in RFC 5011 i.e. 30 days, but can be signalled by the zone owner with the TTL of the TALINK RRset at the zone apex. This TTL MUST be validated before it is stored for use later.

If a validator is also using [RFC5011] for the target zone, then the trust history algorithm SHOULD only be invoked if the last [RFC5011] successful probe was more than 30 days ago. If a new key has been announced, invoke the history if no 2 probes succeeded during the add hold-down time and there was no successful probe after the add hold-down time passed. Therefore the time of the last successful probe MUST be stored on stable storage.

For testing the potentially very infrequently used lookup, the following SHOULD be implemented. For the test the lookup is triggered manually by allowing the system to be given a particular keyset with a last successful lookup date in the past, stored TALINK TTL and a test History Provider. The test History Provider provides access to a generated back-dated test history that signs the current production keyset.

7. Security Considerations

The History Provider only provides copies of old data. If that historic data is altered or withheld the lookup algorithm fails because of validation errors in Step 3 of the algorithm. If the History provider or a Man in the Middle Adversary (MIMA) has access to the original private keys (through theft, cryptoanalisis, or otherwise), history can be altered without failure of the algorithm. Below we only consider MIMAs and assume the History Provider is a trusted party.

Spoofing by a MIMA of data looked up in step 2 or 3, i.e. spoofing of TALINK and DNSKEY data, can present some alternate history. However the DNSKEY RR set trusted that the history should arrive at is already fixed by step 1. If an attempt is made to subvert the algorithm at step 2 or 3, then the result keyset can not be replaced by another keyset unnoticed.

To change the keyset trusted as the outcome, the step 1 data has to be spoofed and the key held by the validator (or a newer historic key) has to be compromised. Unless such spoof is targeted to a specific victim, a spoof of the step 1 result has a high visibility. Since most of the validators that receive the spoof have an up-to-date trust anchor most validators that would receive this spoof return validation failure for data from the zone that contains the DNSKEYs. An adversary will therefore have to target the attack to validators that are in the process of an update. Since validators do

not announce that they use trust history lookup until step 2 adversaries will not be able to select the validators.

A spoof of data in steps 2 and 3, together with a compromised (old) key, can result in a downgrade. At steps 2 and 3 a faked trust point deletion or algorithm rollover can be inserted in a fake history. This avoids the high visibility of spoofing the current key (see previous paragraph) and downgrades to insecure.

Finally there is the case that one of the keys published by the History Providers has been compromised. Since someone spoofing at step 1 of the lookup algorithm and presenting some fake history to a compromised key, of course does not include key revocations and does extend the history to contain the compromised key, it therefore is not really useful for a History Provider to remove the key from the published history. That only makes lookups fail for those validators who are not under attack. Useful action would be to update validators using some other means.

Rollover with [RFC5011] revokes keys after use. If a History Provider is used, then such revoked keys SHOULD be used to perform history tracking and history lookup.

The SEP bit is checked to make sure that control over the KSK is necessary to change the keyset for the target zone.

8. IANA Considerations

Resource record type TALINK has been defined using RFC5395 type expert review, it has RR type number TBD (decimal).

9. Acknowledgments

Thanks to the people who provided review and suggestions, Edward Lewis, Michael StJohns, Bert Hubert, Mark Andrews, Ted Lemon, Steve Crocker, Bill Manning, Eric Osterweil, Wolfgang Nagele, Alfred Hoenes, Olafur Gudmundsson, Roy Arends and Matthijs Mekking.

10. References

10.1. Informative References

[NIST800-57] Barker, E., Barker, W., Burr, W., Polk, W., and M. Smid, "Recommendations for Key Management", NIST SP 800-57, March 2007.

[RFC5011] StJohns, M., "Automated Updates of DNS Security (DNSSEC) Trust Anchors", RFC 5011, September 2007.

10.2. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3597] Gustafsson, A., "Handling of Unknown DNS Resource Record (RR) Types", RFC 3597, September 2003.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

Authors' Addresses

Wouter Wijngaards
NLnet Labs
Science Park 140
Amsterdam 1098 XG
The Netherlands

EMail: wouter@nlnetlabs.nl

Olaf Kolkman
NLnet Labs
Science Park 140
Amsterdam 1098 XG
The Netherlands

EMail: olaf@nlnetlabs.nl