DNSOP WG
INTERNET DRAFT
Category: I-D

Edward Lewis
NAI Labs
November 24,2000

Handling of DNS Zone Signing Keys <draft-ietf-dnsop-keyhand-03.txt>

Status of this Memo

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

This draft expires on May 24, 2001.

Abstract

DNS Security Extensions require a greater interaction between zone administrations sharing a zone cut. The center of the interaction is the handling of the zone keys of the child and the signature applied by the parent. DNSSEC does not include a protocol for this, but the means of this interaction need definition to maintain the security of DNS.

1 Introduction

DNS Security Extensions require a greater interaction between zone administrations sharing a zone cut. The center of the interaction is the handling of the zone keys of the child and the signature applied by the parent. DNSSEC does not include a protocol for this, but the means of this interaction need definition to maintain the security of DNS.

The abstract having been repeated, the following caveats should be stated. This document is a work in progress. The draft is revised in response to lessons learned at various workshops held throughout the past two years. The workshops have been based on ISC's BIND versions **8 and 9.**

Section 2 of the document discusses the issues surrounding keys as they progress from generation to disposal. Section 3 discusses the interactions between a parent and child zone. Section 4 lists requirements for a protocol to carry out the steps described in Section 3.

Note: I appologize for not being able to include most of the recent workshop results. Job duties have derailed me, but I am not able to return my attention to this effort. I will have a more extensively updated draft submitted after the San Diego meetings.

1.1 On-tree Validation

"On-tree validation" refers to the signing of a child zone's (apex) KEY RR set by the parent zone's private key. The child zone would then publish the KEY RR set and the SIG (KEY) RR as generated by the parent. (See note in section 1.1.1.) This makes possible building a chain of trust by a resolver that is verifying an answer it received. The term "on-tree" refers to following the DNS domain hierarchy to reach a trusted key, presumably the root key if no other key is available. The term "validation" refers to the digital signature by the parent to prove the integrity, authentication and authorization of the child' key to sign the child's zone data.

The term "off-tree validation" refers to the use of some domain name other than the parent zone to sign a child's KEY RR set. This ability has been suggested to be a desirable feature from time to time. However, securely accomodating off-tree validation is a hard problem.

So, for this document, on-tree validation is assumed to be the only recognized model. More liberal validation models require more study. Starting with a more conservative model makes transitioning to more liberal models easier, the vice versa is not easy.

1.1.1 Where Are Keys Published? (only really new material in -03)

During the development of DNSSEC, it has often been assumed that, given a secured parent and child zone, the child would publish it's keys and the parent would have the option to do so also. Later documents discouraged the parent zone from publishing the child's keys becuase of the chance of a conflict.

Recent work at NLnet Labs (I hope this is the correct atribution) has suggested that perhaps the child shouldn't publish the keys and that the parent should. The reason for this suggestion was the observation

of the sequence of events when a parent changes it's keys, resulting in a need to revalidate all child keys.

If the parent is publishing the child's (public) keys, then the revalidation can happen smoothly, without the child having to be involved. However, the issue of the parent's liability in holding the child's key is raised. What if the child needs to change their keys suddenly, but the parent is not able to change what is published, resulting in a loss of service to the client?

2 Life Cycle of a Key Pair

A key pair's lifetime begins with its generation and after a period of time the pair is disposed of. The reason for time-limiting keys is that, given enough time, any secret can become exposed, guessed or otherwise "broken." The challenge is to use a key pair for less time than is needed to "break" or "discover" its private component.

There are three primary factors which determine the span of time a key pair is useable. One is the quality of the generation process. Truly random number generation and hard to derive numbers increase the useful life of a key pair. Another factor is the safety of the secret. The less is it used or placed in a vulnerable position, the longer it will be useful. Yet another factor in the so-called strength of the key pair is length, generally, the longer or bigger a key, the longer it takes to break or guess it.

2.1 Key Generation

Generating a key is a simple operation with a proper tool, and DNSSEC tools are available for this. At this time, a key pair is requested that meets the definition of signing zone key pair. Issues for an administraitor at this this stage are the cryptographic algorithm and length of the keys.

DNSSEC has four kinds of algorithms. One is mandatory-to-implement, which means that all DNSSEC compliant software (servers, resolvers) will be able to process the signature. (DSA is in this category.) Another category is standard, which means that there is a definition for the algorithm's use in DNSSEC, but for some reason there may be software that is unable to implement it. (RSA is in this category.) Oddly, the most efficient algorithm for DNSSEC is in the latter category.

The other two categories are not options at this point. One is shared secret (HMAC-MD5) and the other category includes undefined algorithms.

The meaning of the length of a key varies by algorithm. In general, the longer the key, the safer the pair is. However, as safety increases, speed of signing and verification drops. An administratot

should weigh more frequent key generations and deletions against slower name resolutions when choosing a key length.

2.2 Public Key Validation

Before a zone key pair can be useful, it must be validated by the parent zone. To be validated, the public key must be converted into a KEY RR, complete with a fully qualified owner name, time to live value and other settings. The new KEY RR has to be combined with any other KEY RR's comprising the key set owned by the zone's apex. Individual keys are not validated in DNSSEC, entire key sets are validated.

To be validated, all of the keys to be held at the zone's apex are collected and send to the parent. The information sent is specificied in section 4, for now, consider the information to be equivalent to what would be in the on-the-wire format as send in response to a query.

Until the response is received from the parent, the private key of the pair may be put into use, but any signatures resulting from the use will not be useful in DNSSEC validation. (They will be considered to be extra-DNSSEC or immaterial.)

2.3 Signing With Private Key

A private key can be used to sign a zone after generation in parallel with getting the public key validated. But the signatures won't be of use until there is a validating signature over the KEY RR holding the public key.

There are potentially three ways in which a zone private key can be put to use. One is to use the key in a non-server, or off-line, signer application. Another is to place the key in a dynamic update enabled server. Yet another, less common use, is in a trusted dynamic update client that an administrator trusts to properly sign data.

2.3.1 Non-Server Signing

A non-server, signer application is a software process that accepts a zone master file and signs the data in it with one or more private keys, and produces a signed master file. In past DNSSEC documents, the connectivity of a machine running the process is significant.

If the signer machine is not accessible to the network on which the data is served, the machine is said to be "off-line." The purpose for this is enhanced security of the private key. A break-in compromising the server holding the key is less likely if there is no way to remotely send even just IP traffic to it.

Off-line signing is not a requirement for DNSSEC. An on-line signer is permitted, but then host security must be tightly maintained as the a compromise of crytptographic data is often hard to recover. As

opposed to the destruction of some files that could be recovered from backups, exposed cryptographic data could, in the case of DNS, put the zone's data, and all delegated zone's data, at risk of being hijacked.

Note that if off-line signing is practiced, the generation of keys should also be done on this machine, or another off-line machine.

2.3.2 Dynamic Update Signing

A private key is needed in dynamic update to sign, at the least, new SOA records and NXT records. Updated (changed) data sets must be signed too. Dynamic update servers should be careful to keep the private key secret, including locking it into memory and practice good host security measures. The key must not be left in an exposed configuration file on disk unless the file is properly encrypted.

The same comments apply to any dynamic update client entrusted to supply signed data. A wise system administrator will rarely make use of this feature.

2.4 Loading A New Zone

Assuming the intent of putting a zone key into use is to make the zone secure to the world, the validated set of KEY RRs holding all the public keys belonging to the zone's apex must be included in the zone file. If a validating SIG (KEY) RR is not available, then the key set, and all sets signed in the zone will be considered unsecured to all but zones with preconfigured keys.

2.5 Removing a Key From Use

Removing a key from use is as simple as removing the KEY RR pertaining to the public key from DNS. This renders all SIG RR's generated by the private key meaningless.

There are a few considerations when it comes to removing the KEY RR though. Unless the removal of the KEY RR is in response to a security emergency (such as a suspected loss of secrecy of the private key), the private key's use should be terminated first. Cache's may hold SIG RR's up to the TTL, so usually, the KEY RR won't be pulled until the TTL time after the private key is pulled and ideally replaced.

One other consideration is the impact that the removing the KEY RR is the impact on any delegated zones. If the KEY RR is used to validate child zone's keys, then as soon as the KEY RR is removed, the lower zones are no longer properly secured.

2.6 Steps in Performing a Key Rollover

<<This is where I will include the text based upon the meetings at NIC-SE in September.>>

3 Parent-Child Interactions

As stated in the Introduction, on-tree validation is assumed, so all zone key validation will happen between parent and child.

3.1 New Delegation

At the inception of a new zone, besides the traditional data exchange, the child should request validation for an initial set of zone keys. This action would be identical to that described in the next section, except that there may not be an initial NXT to modify.

Since this is the first set of keys used by the child, the parent needs to be sure that the child is truly who they claim to be. There must be some out of band means used to authenticate the new zone administrator. Successive key sets can be installed using the first set's signature, so this authentication is a one-time but crucial step.

The zone administrations should also make plans to handle "stolen" child keys. If a child zone's private key is exposed or stolen, the zone must be able to install new keys and have the parent sign them. The parent must authorize the child again to prevent zone jacking.

<<This is where more of a discussion on child-parent authentication of the key exchange process is discussed, an issue raised at the NIC-SE and NSI/Nanog workshop.>>

3.2 Child Initiated Validation

The most efficient way to accomplish the validation of a zone is to require the child zone to make a request to the parent. This could be in response to the child's decision to change keys, to replace a broken key, or a notification from the parent that the parent has changed its keys.

The child first has to assemble the entire set of KEY RR's that needs validation. The set must be fully specified, meaning that all fields in the RR must be supplied. In particular, the TTL's must be set, this has proven to be a problem in workshops.

The child needs to decide the span of time over which the set is to be validated. A child need not wait until a set of keys is needed "now" to make a request. Consider the following example.

A child zone decides to have a different zone key for each month of the upcoming year. The child also decides to publish just three of the keys at any one time - the key for the previous month (to validate data still being tossed around), the current key, and the key for the next month (in case the KEY set is cached somewhere beyond the end of the month). The child would then assemble 12 KEY RR sets, consisting of three of the keys at a time, and submit them for validation, one

set per month.

The child should also request the parent sign the keys using a set of particular algorithms. If the parent does not recognize any member of the set, that algorithm cannot be supplied. The parent should not return a key of an algorithm that is not requested.

The request is then sent to the parent for validation. The security of this transfer is crucial to the safety of the keys and the entire DNSSEC process. The child must ensure that the request arrives with source integrity. The parent must be able to authenticate the request, that the request is from an authorized source. Non-reputability may be a desired feature.

The parent's processing of the validation must not alter the set of keys submitted by the child. The parent must not add a NULL key if there is no zone key present. If a child submits a key set that has not zone signing KEY in it, then the child zone will remain unsecured. It is not the job of the parent to notice this, this is left to the resolvers.

The parent should not sign any key sets belonging to any member of a child zone other than the apex. Other than the apex key set, the parent should not sign any data in the child zone. It is not the job of the parent to provide security for members of the child zone.

After the SIG (KEY) RR for each KEY RR set to be validated is generated, the result is returned to the child. It is important that the child verify the signatures returned by the parent. If the signatures don't verify the appropriate key set, DNSSEC verification of the zone data will not succeed.

3.3 Parent Initiated Validation

A parent can initiate a validation of a child as a by product of the parent's changing or revoking of its own keys. When a parent removes a key, all children that relied upon the key should be notified that they need to resubmit keys for validation, as decribed in the previous section.

The means a parent uses to inform children is not specified here. Also, whether a parent notifies all children or just children impacted by the change in keys is not specified. The latter issue poses an interesting design decision.

If a parent chooses to notify just the children impacted by the removal of a key, then the parent must retain the knowledge of which key is used to sign which child. As this sounds like an onerous burden, consider that the alternative is to notify all children when any key is removed, and suffer a mass revalidation.

3.4 Removal Of Delegation

When a delegation is ended, data is removed from the parent. But one more step is needed, the keys signing the child zone may have to be removed also. This stops the child data from being authenticated, if the child zone servers are not stopped from answering queries. This is a consideraton only in a non-cooperative removal of a zone.

If keys validating a delegation are removed, a "Parent Initiated Validation" will likely ensue.

3.5 Expiration Of A Validation

When a SIG (KEY) RR generated by the parent for a child's key set expires, there is no requirement that either side act. The child may simply wish to revert to an unsecured state. A parent is under no obligation to make sure that the child zone's are properly operated.

3.6 Other Adjustments

There is a scenario in which a child may want to have the parent indicate that the child does not have a set of validated keys. A child may discover a problem with the key set, such as the loss of the private keys (rm -f *) or the exposure of the keys to an untrusted party. In this case, the parent should have some means for the child to request a change in state.

Note that the reverse transition is not necessarily desirable. A child should not be given the ability to claim that it has validated keys without the parent doing the signing. This draft assumes that on-tree validation is the only permitted model, and this is what drives the comment. Off-tree validation needs much further development before it can be accomodated in a secure manner. When such a model is used, then it might make sense for a child to request being identified as secured without submitting keys for validation.

4 Requirements on the Validation Process

This section has not been complete yet.

4.1 Child Request For Validation

4.2 Parent Response to Child

4.3 Parent Notification Of Key Removal

4.4 Parent Data Retention

5 IANA Considerations

This document does not place any requirements on the assigned numbers authority.

6 Security Considerations

This entire document is a note on security considerations. If the zone key is mishandled, in a way that compromises its security, then the security of its zone is compromised.

7 References

8 Author's Address

Edward Lewis
<lewis@tislabs.com>
3060 Washington Rd (Rte 97)
Glenwood, MD 21738
+1(443)259-2352

9 Acknowledgements

The following individuals and groups have made significant input into the content of this document: the attendees of the NIC-SE work shop on DNSSEC, May 18 and 19, 1999, also Olafur Gudmundsson, and Brian Wellington.

A second workshop held by the CAIRN research network September 29 and 30th also provided input to this document. Dan Massey has provided input based upon this workshop and experience with DNSSEC in CAIRN.

More workshops are to be acknowledged...

10 Full Copyright Statement

NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."