DNSOP WG
INTERNET DRAFT
Catagory: I-D

Edward Lewis
NAI Labs
June 12, 1999

Handling of DNS zone signing keys
<draft-ietf-dnsop-keyhand-00.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet- Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html.

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

This draft expires on December 12, 1999.

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.

This document discusses the issues defining the problems of handling zone signing keys. The document begins with an introduction to keys,

delegations, and security. Following this, a few concepts are described that will have an impact on the discussion to follow. The heart of the document discusses how a key is handled during its lifetime and how keys are transferred between organizations. Finally, some of the surrounding issues are discussed in more detail.

This document is a strawman of the interaction between a parent and child zone. Comments, clarifications, additions and deletions are welcome.

1.1 Usage of the KEY RR

Before discussing how keys are to be handled, the different kinds of keys in KEY RRs and the different roles keys play must be introduced.

Throughout this document, a key is either the pair of asymmetric keys or one of the pair. Symmetric keys are not able to be zone signing keys. A "zone key" refers to the public key of a private-public pair.

The phrase "the zone keys signs data" is used throughout the document. This should be read as "the private key corresponding to a public key marked as a zone key signs data." In subsequent editions of this document, the wording will become more accurate.

1.1.1 Zone keys signing DNS contents

The key of primary interest is the zone key. A zone key is stored at the apex of the zone and has the zone key bit flag on. Zone keys can only be in the apex name of a zone. In order to be valid, the zone key must be signed by an appropriate authority, most commonly the parent.

Policy governing the authentication of data in general, and keys in particular is not well defined. This document will stay neutral on the issue also, so defining the validity of a key is not further refined.

1.1.2 Non-zone keys signing DNS contents

A non-zone key may be allowed to sign some data in the zone. The range of names that such a key may sign is limited in scope compared to a zone key. The non-zone key must be signed by a zone key and have signing flags and key strength bits set accordingly. Such a key may be most useful with dynamic update.

All signing keys must have their flag bits set to allow authentication, and the protocol field set to either DNSSEC or ANY. This applies to zone keys as well.

1.1.3 Keys not used in signing DNS data

DNS can hold keys that are not destined for signing DNS data. A zone key may be restricted from signing data, or simply be no longer in use. Null keys are used to signify that data is not signed. Host and

user keys may be used strictly for email, IPSEC, and other protocols. Apex names, which hold the zone keys, may also hold other keys not signified as zone keys. And finally, the CERT RR holds keys within certificates which are not intended for DNS use.

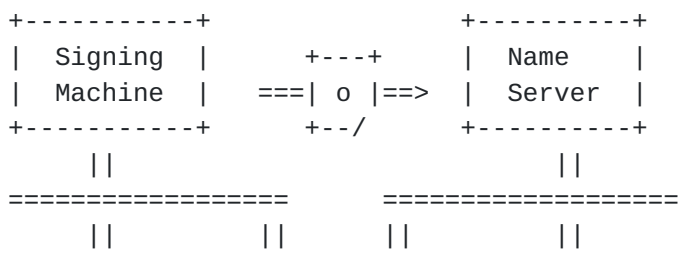
With the exception of keys at the apex, keys not used for signing are of no interest to this document. Non-signing keys at the apex are only a consideration in that they are validated by the parent just as the signing zone keys are validated.

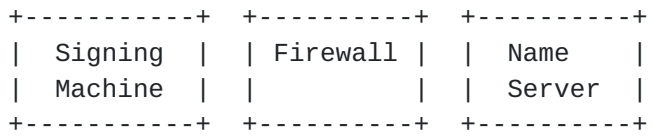
1.2 Off-line and on-line operations

The zone key advertised by a KEY RR is the public key of a public-private key pair. The public key is used to verify data. The private key is used to sign data. Asymmetric key operations are founded on the fact that, given the public key, the private key is not derivable, and only the private key can generate the signature that the public key validates.

The important idea here is that the private key must be protected. Once the private key is exposed or stolen, the public-private key pair has no usefulness. The best way to keep the private key from being exposed or stolen is to keep it away from the network that is using the DNS.

This can be interpreted as a requirement to perform signing on a computer that is not connected to the network, moving data manually (removable media). If this is done, the operator must also be a person trusted not to otherwise make the private key known. This is known as "off-net" signing.





The signing machine need not be isolated from all networks, it may be on an administrative network, as shown above. This "back office" LAN may have an air-gap to the main network (internet or intranet), or there may be a firewall in place (as shown above).

If there is an air gap, then there is an issue of how to move data across it. The most obvious means is some sort of removeable media. If we assume the staff running this part of the network is trustworthy, we don't need to consider deeper issues - such as classified data.

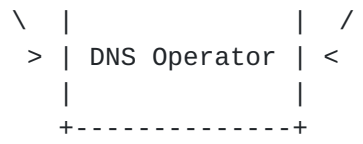
Finally, this off-net signing is a recommendation, not a mandatory configuration. Off-net signing makes the private key harder to expose and/or steal. Performing on-line signing means that the signing machine must be hardened against attack, or the private key is of little value.

Unlike most servers, the compromise of a host with (any kind of) a private key cannot be simply fixed by reinstalling from clean back up tapes and patching the hole. The loss of the key may require new keys to be made, and the revocation of what rights and privileges were tied to the old key.

1.3 Delegation Relationships

Throughout DNS, relationships among the administrations at a delegation point vary in closeness. Between a TLD and a second level domain, a relationship is formal and distant. Further down in the domain structure, a lone DNS administrator may delegate a zone that is also run by the same person.

The terms used to label the range of relationships are inter-organizational, intra-organizational, and intra-office or personal. Naturally, the formality of the operations described in this document will scale accordingly throughout the range of relationships.



There is one other aspect of relationships to introduce. This is the relationship of a DNS registry, the regitrar, the registrant and the operator of the DNS. These four blocks may be distinct, or any pair, trio, or all four, may be combined into one organization.

To preserve generality, this document will consider the inter-organizational relationships and assume that the registry, registrar, registrant, and operator are distinct.

1.4 Minimum Expectation of Security

Setting the expectation of security of DNSSEC is important. A DNSSEC signature means that the data received came from the indicated zone, and not much else. The data is neither necessarily correct, nor even out of date. The data may have arrived from a cache which is holding data whose TTL was set too long by the authoritative owner.

To further refine "how secure" an answer is, a document on signature policy is required. One cannot assume that the signing zone employes strong keys and secure practices to protect its signing operations. Such a document can provide a basis for zone administrators to judge whether their operations are secure "enough."

1.5 Security Considerations

The handling of keys in DNSSEC is not as sensitive as keys used in encryption of sensitive data. Uncompromised keys can be used to begin the use of newer keys, as perfect forward secrecy is not an issue.

Perfect forward secrecy refers to the inability of an attacker to read an ongoing stream of encrypted traffic which changes keys once the attacker has defeated a particular key.

2 Preliminary Digressions

A few issues need to be introduced and discussed. Since these are not directly related to the handling of keys, only a brief discussion is present here. Each of these issues appear in a later section, when the elaboration is more appropriate.

2.1 Dynamic Update

Dynamic update is a recent addition to the DNS operating concept. A zone's contents may be altered on the fly, without reloading the zone.

The impact of this on key handling is that new entries will require new signatures, which will be made using keys available to the Internet. This issue is a dilemma for securing the private key.

2.2 NXT records

NXT records are used to indicate what data is not present in a zone. In particular, each name in the zone's NXT record has the name of the next domain in the zone. Of interest is the situation found in a TLD, each zone delegated from the TLD will have a record pointing to the next "customer" of the TLD. This is the current specification, but is undergoing review. There are some feeling in the operator community that this is not welcome.

2.3 .PARENT file

The .PARENT file is a mechanism in BIND version 4 and 8 used to transfer data from the parent of a delegation to the child. This is not part of the DNSSEC specification, but has been imposed in BIND, because of the way the software is currently written. With time this may disappear, but in the interim, these files will be used, and unfortunately, may present some problems.

BIND version 9 does not plan to use .PARENT files.

3. Key Handling During Lifetime

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

There are two primary factors which determine the span of time a key is useable. One is the quality of the generation process - truely random number generation and hard to derive numbers make a key safer. The other factor in the so-called strength of the key is its length, generally, the longer or bigger the key, the longer it takes to break it.

3.1 Generation of keys

The first step in the life of a zone (or any) key is the generation of the private and public key pair. For DNSSEC, there are initially two kinds of key pairs available, differing in algorithm. One is the mandatory to implement DSA and the other is the optional RSA algorithm.

DSA is an unemcumbered technology which can be included without cost

into server implementations. The RSA algorithm is patented by RSA, Inc., but there is a license to use it for DNS purposes.

RSA signature verification is much quicker than RSA or DSA signature generation. DSA signature verification is slower than generation. (Signatures are more frequently verified than generated.) A DSA signature is much smaller than either an RSA signature or key. A DSA key is bigger than RSA. (Time and size generalizations are made for stated for similar strength key pairs.)

A name server, such as BIND, should provide its own means for generating keys. BIND includes an executable dnskeygen which creates two files, a private key file and a public key file. The public key file is suitable for inclusion in the zone data file. The private key file should be kept protected as this is the secret needed to secure the zone.

3.2 Submitting key for signing

Before a key is put into use, the public key must be signed by an appropriate authority, in this case, the parent zone. A key is not sent individually, rather, a set of keys that will be advertised together are set for signing.

Starting with a new public key, add it to the collection of keys that will be signed together. The keys should be expressed in the KEY RR format, similar to what the key generator has made.

Since key generators will vary in their output, a zone administrator must ensure, whether manually or automatically, that each key record has the appropriate RDATA values - flags, algorithm, protocol and bits, along with appropriate envelope information - type, class, and TTL. The TTLs must be the same throughout the set.

This KEY RR set is sent to the parent, and is kept locally. The parent must also be told what period of validity to apply to the SIG record, i.e., the start and end times for the signature.

The parent must return the newly generated SIG(KEY) RR, one for each set and each required algorithm the parent uses in signing its data. If the parent uses NXT records (now they are mandatory) then NXT record of the child belonging to the parent zone is also sent, signed. (This transfer may be dropped in the future.)

The child must verify the parent's signature over the keys as they were sent to the parent. Blindly relying on the keys (optionally) returned by the parent, or the signature, makes the zone vulnerable to an attack in which more zone keys are inserted between the local zone the parent's signing machine.

3.3 Use of private key

Once a private key is to be used to sign data, this can be done regardless of whether or not the public key's signing is finished by the parent. E.g., if the zone has produced its first key pair, the public key is sent to the parent for signing. While this is happening the private key is put to use simultaneously.

If the use of the private key, i.e., signing the zone, finished before the public keys is signed by the parent, the newly signed zone must wait. The new signatures are not usable until the new key and its signature is available and signed.

Once a private key is used, it may be reused for as long and as often as desired. The longer a span of time in which the key is used raises the vulnerability of the key to exposure or breaking. The key should also be left off-net for greater safety.

Whenever a zone is signed, the presence of any child keys must be noted. if a new signature is generated for a child's key set, the child should be notified and given the new signature.

3.4 Preparation for publishing the zone

Once the zone is signed, and the signed key is available from the parent, the two are combined. The key is also inserted into the name server configuration files or scripts, so that the zone can be validated during the loading of the data.

3.5 Requesting a new signature of the zone keys

SIG RRs contain a validity period, which limits the span of time the

signature is trusted to verify the integrity of the the data. This introduces the concept of data expiring from even the primary authoritative server.

When the SIG RR covering the zone keys is about to expire, and the parent has not already begun to generate a new signature, the child zone will have to request a new signature from the parent. It is not clear if the child must resend the keys, since the parent should have them already.

3.6 Disposing of a key

Removing a key from the DNS is as simple as erasing it and the signatures it generated. There is no reason to archive old keys as these are simply providing signatures, not encrypted data. There is

no risk in old uses of a key coming back, such as someone unlocking an encrypted message, since there is a fundamental assumption that DNS holds no secret data.

4. Key Transfer between Zone Administrations

There are a number of data flows that are created by DNSSEC to handle keys. This section discusses each starting from the motivation for the flow and includes the required data and operations involved.

4.1 New delegation

At the inception of a new zone, besides the traditional data exchange, the child must give to the parent a set of zone keys to start the security of the zone. If the zone is initially unsigned, then the parent assigns NULL keys to the zone. The parent, prior to installing the new delegation records must supply a signature covering the keys, installing this signature in its zone and sending the SIG RR to the child. See the discussion in the next section (4.2) for more details.

Since this is the first set of keys used by the child, the parent needs to be sure that the child is truely 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 hi-jacking of a zone.

The parent and child must also agree on how to handle the refreshing of signatures. If a child's SIG(KEY) RR is about to become invalid, and the parent hasn't sent a new one, the child must request a new one. Handling "emergency" signings and/or having a fixed request/response period are two operations that should be defined in the agreement covering the delegation.

4.2 New zone key set

When a child creates a new zone key (or keys) intending to publish the key in its zone, it adds the key to the existing keys it plans to continue publishing. E.g., a zone may create one new key pair each month for use, and always retain the keys of the previous two months in the zone. Hence, there are always three keys in the DNS. When keys A, B, and C, are in the zone, and the next month begins, key D is made. A, B, and C are already signed, but now key D replaces key A.

So the set of keys B, C, and D are sent to the parent for signing.

A suggested scenario is that a zone administrator may generate 12 keys in a year, and plan to use one a month. At any one time, the key for the previous month, the current key, and the next month's key are to be advertised. Instead of contacting the parent each month with a new set of keys, the child may send the keys in groups at one time. E.g., for keys named A-L, the first set would be A-C, the next B-D, then C-E, and so on. The validity period for each trio would be one month after the previous trio.

For each key set sent to the parent, the child must supply fully defined resource records. This includes the fully qualified domain name, class, type and time-to-live. The child must also specify the period of validity of the SIG RR.

In addition, while the parent uses NXT records and passes them to the child (the current operation), the child must tell the parent what its zone minimum TTL is, so that the parent can set the TTL on the resulting TTL RR to the minimum of the parent's and child's TTLs. The period of validity on the SIG(NXT) RR should match the SIG(KEY) RR period.

Upon receiving the SIG(RR) and other records from the parent, the child must verify that the new SIG covers the KEY RR set as it was sent - looking for accidently or maliciously inserted keys. This means that the child must be able to find the parent's signing public key. If this check is done on an off-net machine, then the parent's key must be configured as a DNS lookup isn't possible.

4.3 Child requests a signing

A child zone may need a new signature to cover the keys without generating a new key. In this instance, the same data as mentioned in the previous section must flow, perhaps with a notice that this is just a signature refresh operation.

4.4 New parent key

When the parent zone signs with a new key and retires signatures generated with an old key, the new SIG(KEY) and SIG(NXT) (if used) must be send to each of the respective delegations.

The child must first verify the new SIG(KEY) and add it to the zone if

it passes. If the signature does not verify, there may be a problem with the set of keys held for the zone at the parent.

4.5 Zone key exposed

When a zone key is exposed, broken, or stolen, the zone needs to start anew with a new key set. The out of band mechanisms set up at the generation of the zone should be used to get this new set of keys signed.

4.6 Removal of delegation

A parent zone can always remove a child zone by simply erasing the delegation records and adjusting the NXT records pointing to the name. There is no special provision for informing the child of this. However, the parent should be aware that unless the key used to sign the child's keys is removed, the child's data will remain valid until the signatures expire.

5. Further Digressions

Issues that were introduced in section 2 are brought back here for further elaboration.

5.1 Dynamic Update

The care taken to keep a private key from view, namely the use of off-net signing, can not be extended to dynamic update situations. Data added to a secured zone through a dynamic update must be signed just like all the other data in a zone. Therefore, there must be a private key available on line.

There are a few ways to maintain the high level of security of data signed off-net when mixed with the use of dynamic update. One way is to use separate zones for data signed off-net and data changed via updates. By "quarantining" the updates, no update can interfere with the static data.

Another means is to use the strength bits in the key flags to prevent off-net signed data from being modified by a dynamic update. This issue hasn't been resolved yet, however.

5.2 NXT records

NXT records represent another data flow - the NXT itself may be updated without a change in the KEY or SIG(KEY). This occurs when the parent delegates a zone whose name is lexically immediately after the child's name. The parent must inform the child to replace the NXT record it previously supplied with the new one. The child must be aware that there are two NXTs at the zone apex, corresponding to the parent and child zones, respectively.

At this time, there is some confusion about the impact of this on the SOA and the informing of the secondaries of this change. More will

follow later.

5.3 .PARENT file

The .PARENT file is a mechanism currently used to save data from being removed when loading successive zones in a name server. While this is used, the .PARENT file provides a convienent format for communication from the parent to the child. The parent's response to the child may be just the .PARENT file - assuming fully specified text representations are used. Specifically, the TTL must appear.

The .PARENT file lacks information that the child needs to pass to the parent. What is missing is: protection of the key set from the insertion of other keys (which could be a signature of the set using an older key), the desired validity period, and the child's minimum ttl. This data must be worked into the format sent from child to parent when requesting a new signature. In addition, deadline data may also be required, this depends on the agreement made at the time the zone is delegated.

Once again, BIND version 9 does not use the .PARENT file.

6 IANA Considerations

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

7 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.

8 Author's Address

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

9 Acknowledgements

The following individuals and groups have made signifiant 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.

10 References

This section will be more formally defined as the doument progresses.

RFC 2535 defines DNSSEC
RFC 1035 is the start of the definition of DNS

RFC 2136 defines DNS Dynamic Updates

11 Full Copyright Statement