DNSIND WG   Edward Lewis
INTERNET DRAFT   NAI Labs
Category: I-D

December 25, 1999

DNS Security Extension Clarification on Zone Status
<draft-ietf-dnsind-zone-status-00.txt>

Comments should be sent to the authors or the DNSIND WG mailing list namedroppers@internic.net.

This draft expires on June 25, 1999.

Abstract

The definition of a secured zone is presented, updating RFC 2535. The new definition has consequences which alter the interpretation of the NXT record, obsolete NULL keys, and the designation of "experimentally secure."

1 Introduction

Whether a DNS zone is "secured" or not is a question asked in at least four contexts. A zone administrator asks the question when configuring a zone to use DNSSEC. A dynamic update server asks the question when an update request arrives, which may require DNSSEC processing. A delegating zone asks the question of a child zone when the parent enters data indicating the status the child. A resolver asks the question upon receipt of data belonging to the zone.

A zone administrator needs to be able to determine what steps are needed to make the zone as secure as it can be. Realizing that due to

the distributed nature of DNS and its administration, any single zone is at the mercy of other zones when it comes to the appearance of security. This document will define what makes a zone qualify as secure (absent interaction with other zones).

A name server performing dynamic updates needs to know whether a zone being updated is to have signatures added to the updated data, NXT records applied, and other required processing. In this case, it is conceivable that the name server is configured with the knowledge, but being able to determine the status of a zone by examining the data is a desirable alternative to configuration parameters.

A delegating zone is required to indicate whether a child zone is secured. The reason for this requirement lies in the way in which a resolver makes its own determination about a zone (next paragraph). To shorten a long story, a parent needs to know whether a child should be considered secured. This is a two part question, what does a parent consider a secure child to be, and how does a parent know if the child conforms?

A resolver needs to know if a zone is secured when the resolver is processing data from the zone. Ultimately, a resolver needs to know whether or not to expect a usable signature covering the data. How this determination is done is out of the scope of this document, except that, in some cases, the resolver will need to contact the parent of the zone to see if the parent states that the child is secured.

This document updates several sections of RFC 2535. The definition of a secured zone is an update to section 3.4 of the RFC. The document updates section 2.3.4, by specifying a replacement for the NULL zone keys. The document also updates section 3.4 to eliminate the definition of experimental keys and illustrate a way to still achieve the functionality they were designed to provide.

2 Status of a Zone

In this section, rules governing a zone's DNSSEC status are presented. There are three levels of security defined; full, private, and unsecured. A zone is fully secure when it complies with the strictest set of DNSSEC processing rules. A zone is privately secured when it is configured in such a way that only resolvers that are appropriately configured see the zone as secured. All other zones are unsecured.

Note: there currently is no other document completely defining DNSSEC processing rules. For the purposes of this document, the strictest rules are assumed to state that the verification chain of zone keys parallels the delegation tree up to the root zone. This is not intended to disallow alternate verification paths, just to establish a baseline definition.

To avoid repetition in the rules below, the following term is defined.

2.a. Zone signing KEY RR - A KEY RR whose flag field has the value 01 for name type (indicating a zone key) and either value 00 or value 01 for key type (indicating a key permitted to authenticate data). (See RFC 2535, section 3.1.2). The KEY RR also has a protocol octet value of DNSSEC (3) or All (255).

2.1 Fully Secured

A fully secured zone, in a nutshell, is a zone that uses only mandatory to implement algorithms (RFC 2535, section 3.2) and relies on a key certification chain that parallels the delegation tree. Fully secured zones are defined by the following rules.

2.1.a. The zone's apex MUST have a KEY RR set. There MUST be at least one zone signing KEY RR (2.a) of a mandatory to implement algorithm in the set.

2.1.b. The zone's apex KEY RR set MUST be signed by a private key belonging to the parent zone. The private key's public companion MUST be a zone signing KEY RR (2.a) of a mandatory to implement algorithm and owned by the parent's apex.

If a zone cannot get a conforming signature from the parent zone, the child zone cannot be considered fully secured.

2.1.c. NXT records MUST be deployed throughout the zone. (Updates RFC 2535, section 2.3.2.) Note: there is some operational discomfort with the current NXT record. This requirement is open to modification when two things happen. First, an alternate mechanism to the NXT is defined and second, a means by which a zone can indicate that it is using an alternate method.

2.1.d. Each RR set that qualifies for zone membership MUST be signed by a key that is in the apex's KEY RR set and is a zone signing KEY RR (2.a) of a mandatory to implement algorithm. (Updates 2535, section 2.3.1.)

2.2 Privately Secured

A privately secured zone is a zone that complies with rules like those for fully secured, with the following exceptions. The signing keys may be of an algorithm that is not mandatory to implement and/or the verification of the zone keys in use may rely on a verification chain that is not parallel to the delegation tree.

2.2.a. The zone's apex MUST have a KEY RR set. There MUST be at least one zone signing KEY RR (2.a) in the set.

2.2.b. The zone's apex KEY RR set MUST be signed by a private key and one of the following two sentences MUST hold true. The private key's public companion MUST be preconfigured in all the resolvers of interest. The private key's public component MUST be a zone signing KEY RR (2.a) authorized to provide validation of the zone's apex KEY RR set, as recognized by resolvers of interest.

The previous sentence is trying to convey the notion of using a trusted third party to provide validation of keys. If the domain name owning the validating key is not the parent zone, the domain name must represent someone the resolver trusts to provide validation.

2.2.c. NXT records MUST be deployed throughout the zone. (Updates RFC 2535, section 2.3.2.) Note: see the discussion following 2.1.c.

2.2.d. Each RR set that qualifies for zone membership MUST be signed by a key that is in the apex's KEY RR set and is a zone signing KEY RR (2.a). (Updates 2535, section 2.3.1.)

2.3 Unsecured

All other zones qualify as unsecured. This includes zones that are designed to be experimentally secure, as defined in a later section on that topic.

2.4 Wrap up

The designation of fully secured, privately secured, and unsecured are merely labels to apply to zones, based on their contents. Resolvers, when determining whether a signature is expected or not, will only see a zone as secured or unsecured.

Resolvers that follow the most restrictive DNSSEC verification rules will only see fully secured zones as secured, and all others as unsecured, including zones which are privately secured. Resolvers which are not as restrictive, such as those that implement algorithms in addition to the mandatory to implement algorithms, will see some privately secured zones as secured.

The intent of the labels "fully" and "privately" is to identify the specific attributes of a zone. The words are chosen to assist in the writing of a document recommending the actions a zone administrator take in making use of the DNS security extensions. The words are explicitly not intended to convey a state of compliance with DNS security standards.

3 Parental Notification

For a resolver to come to a definitive answer concerning a zone's security status, there is a requirement that the parent of a zone signify whether the child zone is signed or not. The justification of this requirement requires a discussion of the resolver's activity, which is described in RFC 2535.

In RFC 2535, a parent is required to hold a NULL key for an unsigned child (a bone of contention here is how this works in light of multiple algorithms). The parent has the option to hold the keys of the child if the child is signed. The parent may also hold nothing cryptographic if the child is signed. This, of course, assumes the parent is a signed zone.

There is a strong case for discouraging a parent from holding keys of a signed child. These include concrete concerns about performance and more abstract concerns about the liability of the parent.

DNS [RFC 1034 and 1035] requires a parent to hold NS records for a child zone, this signifies the delegation. RFC 2535 requires a secured parent to also have signed NXT records for the child, and possibly a signed KEY RR set (required for NULL key situations).

By redefining the security status of a zone to be per zone and not per algorithm, there is an opportunity to completely remove the need for a KEY RR set in the parent. Because the question of whether the zone is secure or not is a yes-or-no question, the notification needs just one bit to be expressed.

Keep in mind that the following sections speak to the contents of a zone, not a name server. In the case of a name server speaking authoritatively for both the parent and child, or if a server caches the information for the other half of the delegation, then a server will have more types of data at a delegation point than a parent is supposed to hold. (E.g., if a parent zone's name server caches the SOA for the child, the SOA is not in the parent zone, but is in the server's cache.)

3.1 Child Is Secured Bit

This section is written assuming the current definition of NXT holds. There is some controversy surrounding the NXT record which may result in a complete replacement of it for proof of non-existence. The current NXT definition provides an extension bit in the types present bit map, whose use is remains incompletely defined. The following text largely ignores these uncertainties, and should be rewritten to accommodate these uncertainties in revisions.

In the parent's half of the delegation point, there will be an NXT record. According to the rules for a delegation point, only the NS, NXT, and SIG bits will be turned on in the types present field, assuming we drop the KEY set altogether.

The KEY bit in the parent's NXT types present bit map is hereby redefined to have the following meaning.

If the bit corresponding to the KEY RR set in a parent NXT is set, the parent has signed a KEY RR set for the child that includes a zone signing KEY RR (2.a). Furthermore, the validity period on the SIG (KEY) RR covers the current time and the public component of the key used to generate the SIG (KEY) RR is validly available from the parent.

E.g., Assume the zone "test." signs a key for "zone1.test.," with the signature valid from May 1st to June 1st and a public key from "test." available from April 1st until July 1st. The NXT record for "zone1.test." will have the KEY RR bit set from May 1st to June 1st.

This constraint may be enforced in the SIG (NXT) RR validity period, timely editing of the master file, or whatever other mechanism "test." chooses to implement.

Conversely, if the bit is 0, then the child is not secured. Note that for a fully secured zone (section 2.1), the bit is on (1). For all unsecured zones (section 2.3) the bit is off (0). For privately secured zones (section 2.2), the setting of the bit is determined by whether the parent signs the child's keys or not. Hence, for privately secured zones, the parent may have no responsibility. A child wishing to have the parent set the bit must contact the parent.

3.2 Rules Governing the Bit

In this section, the words of the previous are turned into definitive MUST and SHOULDs. Note that this section does not refer to the bit in the NXT. This is in anticipation of a change in the way NXT indicates types present (e.g., if bit 0 of the field is defined) or a successor to the NXT is defined.

3.2.a. At a delegation point, a parent zone MUST have a mechanism in place to indicate which RR sets are present. The mechanism MUST indicate that the NS, SIG, and the type(s) corresponding to the mechanism itself are present (of course, with these types actually being present). With the exception of the KEY RR type, all other types MUST be indicated as not present, and, in accordance with delegation rules, actually be absent from the zone. If, in the future, other data is permitted to be present at a delegation point, this requirement MUST be amended.

Assuming the NXT record, the above requirement reads as follows. At a delegation point, a parent zone must have a secured NXT record. This NXT record must indicate that the NS, SIG, and NXT types are present. With the exception of KEY, all other types must be indicated as not present. The lower casing of the word "must" is intentional, conveying that this is an explanation of the rule above.

3.2.b. The KEY set MUST be indicated as present during the time when the parent has issued a signature for the child's KEY set. Conversely, during periods of time in which the parent knows it has not generated a signature for the KEY RR set, the KEY set MUST be indicated to be absent.

If the parent has issued signatures with discontinuous validity spans, then the presence of the KEY set will flip from present to not present and back as time progresses.

If the parent is aware that the child's keys are becoming valid or will be becoming invalid at a certain point in time, it is recommended that this be reflected in the NXT's signature validity period.

3.2.c. When signing a child's KEY RR set, a parent SHOULD carefully consider the algorithm of the key used to generate the signature. The parent SHOULD make clear to child zones what steps are to be taken to

get the parent to indicate that the child is signed. This document will go no further in specifying operational considerations.

(Let's say the parent signs the child's key set with an algorithm the child can't process. The child could elect not to advertise this signature as it cannot verify that the signature covers the real key set. If this happens, is the parent justified in claiming that the child is secured?)

3.2.d. The parent MUST allow the child, through some trusted, probably non-DNS mechanism, to request that the indication of the KEY set in the NXT be turned off. This allows a child to revert to an unsigned state.

3.2.e. The parent SHOULD NOT allow the child to request that the KEY set be indicated in the absence of a key signing request.

3.3 Operational Considerations

Retrieving the NXT for a delegated name from the parent zone (the upper NXT) is not a trivial operation. The complication arises due to having an NXT in the delegatee (the lower NXT) that matches the owner name of the upper NXT. (The case in which both the parent and child zones are secured is the only case mentioned here. If both are not secured, then there will be at most one NXT, which is easily retrieved.)

There are two complications to describe. One involves the multiple NXT sets matching the same owner. The other is the pragmatic issue of knowing where to get the answer.

With multiple NXT sets at the same owner, caches may become a problem. If a (for example) recursive server has cached the lower NXT, any query for the upper NXT may be confused for a lower NXT query. This is akin to the issue of the ANY query, where a server with some cached data will answer with just that and not seek the rest of the data.

A resolver may know the child's server's addresses and the parent zone may not be sharing servers with the child. In this case the resolver will need to be able to locate the parent zones (possibly having to go to the root servers and descend) in order to obtain the upper NXT record.

A potential solution to this is to define an NXT meta-query which will force a recursive server to find all available NXT RR sets for a given name. Details of this have not been examined.

4 NULL keys

Through the use of the types present to indicate the existence of a signature validating the KEY set of a child, the need for NULL keys effectively disappears. NULL keys are left as a defined entity, but are rendered meaningless in DNSSEC.

5 Experimental Status

Without NULL keys, an experimentally secured zone cannot be defined as it is in RFC 2535. The purpose of an experimentally secured zone was to facilitate the migration from an unsecured zone to a secured zone.

The objective of facilitating the migration can be achieved without a special designation of an experimentally secure status.

Experimentally secured is a special case of privately secured. A zone administrator can achieve this by publishing a zone with signatures and configuring a set of test resolvers with the corresponding public keys. Even if the public key is published in a KEY RR, as long as there is no parent signature, the resolvers will need some preconfiguration to know to process the signatures. This allows a zone to be secured with in the sphere of the experiment, yet still be registered as unsecured in the general Internet.

6 IANA/ICANN Considerations

This document does not request any action from an assigned number authority nor recommends any actions.

7 Security Considerations

Without a means to enforce compliance with specified protocols or recommended actions, declaring a DNS zone to be "completely" secured is impossible. Even if, assuming an omnipotent view of DNS, one can declare a zone to be properly configured for security, and all of the zones up to the root too, a misbehaving resolver could be duped into believing bad data. If a zone and resolver comply, a non-compliant or subverted parent could interrupt operations. The best that can be hoped for is that all parties are prepared to be judged secure and that security incidents can be traced to the cause in short order.

8 Acknowledgements

The need to refine the definition of a secured zone has become apparent through the efforts of the participants at two DNSSEC workshops, sponsored by the NIC-SE (.se registrar) and CAIRN (a DARPA-funded research network). Further discussions leading to the document include Olafur Gudmundsson, Russ Mundy, Robert Watson, and Brian Wellington.

9 References

[RFC1034] P. Mockapetris, "Domain Names - Concepts and Facilities," RFC 1034, November 1987.

[RFC1035] P. Mockapetris, "Domain Names - Implementation and Specification," RFC 1034, November 1987.

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

[RFC2136] P. Vixie (Ed.), S. Thomson, Y. Rekhter, J. Bound "Dynamic Updates in the Domain Name System," RFC 2136, April 1997.

[RFC2535] D. Eastlake, "Domain Name System Security Extensions," RFC 2535, March 1999.

10 Author Information

Edward Lewis
NAI Labs
3060 Washington Road
Glenwood, MD 21738
+1 443 259 2352
<lewis@tislabs.com>

11 Full Copyright Statement