Network Working Group                                        E. Lewis
Request for Comments: 3090                                   NAI Labs
Category: Standards Track                                  March 2001


            DNS Security Extension Clarification on Zone Status

Status of this Memo

Copyright Notice

Abstract

   The definition of a secured zone is presented, clarifying and
   updating sections of RFC 2535.  RFC 2535 defines a zone to be secured
   based on a per algorithm basis, e.g., a zone can be secured with RSA
   keys, and not secured with DSA keys.  This document changes this to
   define a zone to be secured or not secured regardless of the key
   algorithm used (or not used).  To further simplify the determination
   of a zone's status, "experimentally secure" status is deprecated.

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.

1.1 When a Zone's Status is Important

   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.

   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.  Under
   what circumstances does a parent consider a child zone to be secure,
   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.

1.2 Islands of Security

   The goal of DNSSEC is to have each zone secured, from the root zone
   and the top-level domains down the hierarchy to the leaf zones.
   Transitioning from an unsecured DNS, as we have now, to a fully
   secured - or "as much as will be secured" - tree will take some time.
   During this time, DNSSEC will be applied in various locations in the
   tree, not necessarily "top down."

   For example, at a particular instant, the root zone and the "test."
   TLD might be secured, but region1.test. might not be.  (For
   reference, let's assume that region2.test. is secured.)  However,
   subarea1.region1.test. may have gone through the process of becoming
   secured, along with its delegations.  The dilemma here is that
   subarea1 cannot get its zone keys properly signed as its parent zone,
   region1, is not secured.

   The colloquial phrase describing the collection of contiguous secured
   zones at or below subarea1.region1.test. is an "island of security."
   The only way in which a DNSSEC resolver will come to trust any data
   from this island is if the resolver is pre-configured with the zone
   key(s) for subarea1.region1.test., i.e., the root of the island of
   security.  Other resolvers (not so configured) will recognize this
   island as unsecured.

An island of security begins with one zone whose public key is pre-
configured in resolvers.  Within this island are subzones which are
also secured.  The "bottom" of the island is defined by delegations
to unsecured zones.  One island may also be on top of another -
meaning that there is at least one unsecured zone between the bottom
of the upper island and the root of the lower secured island.

Although both subarea1.region1.test. and region2.test. have both been
properly brought to a secured state by the administering staff, only
the latter of the two is actually "globally" secured - in the sense
that all DNSSEC resolvers can and will verify its data.  The former,
subarea1, will be seen as secured by a subset of those resolvers,
just those appropriately configured.  This document refers to such
zones as being "locally" secured.

In RFC 2535, there is a provision for "certification authorities,"
entities that will sign public keys for zones such as subarea1.
There is another document, [RFC3008], that restricts this activity.
Regardless of the other document, resolvers would still need proper
configuration to be able to use the certification authority to verify
the data for the subarea1 island.

1.2.1 Determining the closest security root

Given a domain, in order to determine whether it is secure or not,
the first step is to determine the closest security root.  The
closest security root is the top of an island of security whose name
has the most matching (in order from the root) right-most labels to
the given domain.

For example, given a name "sub.domain.testing.signed.exp.test.", and
given the secure roots "exp.test.", "testing.signed.exp.test." and
"not-the-same.xy.", the middle one is the closest.  The first secure
root shares 2 labels, the middle 4, and the last 0.

The reason why the closest is desired is to eliminate false senses of
insecurity because of a NULL key.  Continuing with the example, the
reason both "testing..." and "exp.test." are listed as secure root is
presumably because "signed.exp.test." is unsecured (has a NULL key).
If we started to descend from "exp.test." to our given domain
(sub...), we would encounter a NULL key and conclude that sub... was
unsigned.  However, if we descend from "testing..." and find keys
"domain...." then we can conclude that "sub..." is secured.

Note that this example assumes one-label deep zones, and assumes that
we do not configure overlapping islands of security.  To be clear,
the definition given should exclude "short.xy.test." from being a
closest security root for "short.xy." even though 2 labels match.

Overlapping islands of security introduce no conceptually interesting ideas and do not impact the protocol in anyway.  However, protocol implementers are advised to make sure their code is not thrown for a loop by overlaps.  Overlaps are sure to be configuration problems as islands of security grow to encompass larger regions of the name space.

1.3 Parent Statement of Child Security

In 1.1 of this document, there is the comment "the parent states that the child is secured."  This has caused quite a bit of confusion.

The need to have the parent "state" the status of a child is derived from the following observation.  If you are looking to see if an answer is secured, that it comes from an "island of security" and is properly signed, you must begin at the (appropriate) root of the island of security.

To find the answer you are inspecting, you may have to descend through zones within the island of security.  Beginning with the trusted root of the island, you descend into the next zone down.  As you trust the upper zone, you need to get data from it about the next zone down, otherwise there is a vulnerable point in which a zone can be hijacked. When or if you reach a point of traversing from a secured zone to an unsecured zone, you have left the island of security and should conclude that the answer is unsecured.

However, in RFC 2535, section 2.3.4, these words seem to conflict with the need to have the parent "state" something about a child:

    There MUST be a zone KEY RR, signed by its superzone, for every subzone if the superzone is secure.  This will normally appear in the subzone and may also be included in the superzone.  But, in the case of an unsecured subzone which can not or will not be modified to add any security RRs, a KEY declaring the subzone to be unsecured MUST appear with the superzone signature in the superzone, if the superzone is secure.

The confusion here is that in RFC 2535, a secured parent states that a child is secured by SAYING NOTHING ("may also be" as opposed to "MUST also be").  This is counter intuitive, the fact that an absence of data means something is "secured."  This notion, while acceptable in a theoretic setting has met with some discomfort in an operation setting.  However, the use of "silence" to state something does indeed work in this case, so there hasn't been sufficient need demonstrated to change the definition.

1.4 Impact on RFC 2535

   This document updates sections of RFC 2535.  The definition of a
   secured zone is an update to section 3.4 of the RFC.  Section 3.4 is
   updated to eliminate the definition of experimental keys and
   illustrate a way to still achieve the functionality they were
   designed to provide.  Section 3.1.3 is updated by the specifying the
   value of the protocol octet in a zone key.

1.5 "MUST" and other key words

   The key words "MUST", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY"
   in this document are to be interpreted as described in [RFC 2119].
   Currently, only "MUST" is used in this document.

2 Status of a Zone

   In this section, rules governing a zone's DNSSEC status are
   presented.  There are three levels of security defined: global,
   local, and unsecured.  A zone is globally secure when it complies
   with the strictest set of DNSSEC processing rules.  A zone is locally
   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 document completely defining DNSSEC
   verification 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.  (See 2.b below.)
   This is not intended to disallow alternate verification paths, just
   to establish a baseline definition.

   To avoid repetition in the rules below, the following terms are
   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).

   The definition updates RFC 2535's definition of a zone key.  The
   requirement that the protocol field be either DNSSEC or ALL is a new
   requirement (a change to section 3.1.3.)

   2.b On-tree Validation - The authorization model in which only the
   parent zone is recognized to supply a DNSSEC-meaningful signature
   that is used by a resolver to build a chain of trust from the child's

keys to a recognized root of security.  The term "on-tree" refers to
following the DNS domain hierarchy (upwards) 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's key to
sign the child's zone data.

2.c Off-tree Validation - Any authorization model that permits domain
names other than the parent's to provide a signature over a child's
zone keys that will enable a resolver to trust the keys.

2.1 Globally Secured

A globally 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 (on-
tree validation).  Globally 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 globally secured.  The only exception
to this is the root zone, for which there is no parent zone.

2.1.c. NXT records MUST be deployed throughout the zone.  (Clarifies
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.)

Mentioned earlier, the root zone is a special case.  The root zone
will be considered to be globally secured provided that if conforms
to the rules for locally secured, with the exception that rule 2.1.a.
be also met (mandatory to implement requirement).

2.2 Locally Secured

   The term "locally" stems from the likely hood that the only resolvers
   to be configured for a particular zone will be resolvers "local" to
   an organization.

   A locally secured zone is a zone that complies with rules like those
   for a globally secured zone 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
   (off-tree validation).

   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 subclauses MUST hold true.

   2.2.b.1 The private key's public companion MUST be pre-configured in
   all the resolvers of interest.

   2.2.b.2 The private key's public companion 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.  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 globally secured, locally 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 globally secured zones as secured, and all others as
   unsecured, including zones which are locally secured.  Resolvers that
   are not as restrictive, such as those that implement algorithms in
   addition to the mandatory to implement algorithms, will see some
   locally secured zones as secured.

   The intent of the labels "global" and "local" 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 Experimental Status

   The purpose of an experimentally secured zone is to facilitate the
   migration from an unsecured zone to a secured zone.  This distinction
   is dropped.

   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 locally 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 pre-
   configuration 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.

4 IANA Considerations

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

5 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.

6 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), CAIRN (a DARPA-
   funded research network), and other workshops.  Further discussions
   leading to the document include Olafur Gudmundsson, Russ Mundy,
   Robert Watson, and Brian Wellington.  Roy Arends, Ted Lindgreen and
   others have contributed significant input via the namedroppers
   mailing list.

7 References

   [RFC1034] Mockapetris, P., "Domain Names - Concepts and Facilities",
             STD 13, RFC 1034, November 1987.

   [RFC1035] Mockapetris, P., "Domain Names - Implementation and
             Specification", STD 13, RFC 1035, November 1987.

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

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

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

   [RFC3007] Wellington, B., "Simple Secure Domain Name System (DNS)
             Dynamic Update", RFC 3007, November 2000.

   [RFC3008] Wellington, B., "Domain Name System Security (DNSSEC)
             Signing Authority", RFC 3008, November 2000.

10 Author's Address

   Edward Lewis
   NAI Labs
   3060 Washington Road Glenwood
   MD 21738

   Phone: +1 443 259 2352
   EMail: lewis@tislabs.com

11 Full Copyright Statement

Acknowledgement