                 DNS Security Introduction and Requirements

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Abstract

   The Domain Name System Security Extensions (DNSSEC) add data origin
   authentication and data integrity to the Domain Name System. This
   document introduces these extensions and describes their capabilities
   and limitations. This document also discusses the services that the
   DNS security extensions do and do not provide. Last, this document
   describes the interrelationships between the documents that
   collectively describe DNSSEC.

Table of Contents

1. Introduction

   This document introduces the Domain Name System Security Extensions
   (DNSSEC). This document and its two companion documents ([RFC4034]
   and [RFC4035]) update, clarify, and refine the security extensions
   defined in [RFC2535] and its predecessors. These security extensions
   consist of a set of new resource record types and modifications to
   the existing DNS protocol ([RFC1035]). The new records and protocol
   modifications are not fully described in this document, but are
   described in a family of documents outlined in Section 10. Sections
   3 and 4 describe the capabilities and limitations of the security
   extensions in greater detail. Section 5 discusses the scope of the
   document set. Sections 6, 7, 8, and 9 discuss the effect that these
   security extensions will have on resolvers, stub resolvers, zones,
   and name servers.

   This document and its two companions obsolete [RFC2535], [RFC3008],
   [RFC3090], [RFC3445], [RFC3655], [RFC3658], [RFC3755], [RFC3757], and
   [RFC3845]. This document set also updates but does not obsolete
   [RFC1034], [RFC1035], [RFC2136], [RFC2181], [RFC2308], [RFC3225],
   [RFC3007], [RFC3597], and the portions of [RFC3226] that deal with
   DNSSEC.

The DNS security extensions provide origin authentication and
integrity protection for DNS data, as well as a means of public key
distribution. These extensions do not provide confidentiality.

2. Definitions of Important DNSSEC Terms

This section defines a number of terms used in this document set.
Because this is intended to be useful as a reference while reading
the rest of the document set, first-time readers may wish to skim
this section quickly, read the rest of this document, and then come
back to this section.

Authentication Chain: An alternating sequence of DNS public key
    (DNSKEY) RRsets and Delegation Signer (DS) RRsets forms a chain of
    signed data, with each link in the chain vouching for the next. A
    DNSKEY RR is used to verify the signature covering a DS RR and
    allows the DS RR to be authenticated. The DS RR contains a hash
    of another DNSKEY RR and this new DNSKEY RR is authenticated by
    matching the hash in the DS RR. This new DNSKEY RR in turn
    authenticates another DNSKEY RRset and, in turn, some DNSKEY RR in
    this set may be used to authenticate another DS RR, and so forth
    until the chain finally ends with a DNSKEY RR whose corresponding
    private key signs the desired DNS data. For example, the root
    DNSKEY RRset can be used to authenticate the DS RRset for
    "example." The "example." DS RRset contains a hash that matches
    some "example." DNSKEY, and this DNSKEY's corresponding private
    key signs the "example." DNSKEY RRset. Private key counterparts
    of the "example." DNSKEY RRset sign data records such as
    "www.example." and DS RRs for delegations such as
    "subzone.example."

Authentication Key: A public key that a security-aware resolver has
    verified and can therefore use to authenticate data. A
    security-aware resolver can obtain authentication keys in three
    ways. First, the resolver is generally configured to know about
    at least one public key; this configured data is usually either
    the public key itself or a hash of the public key as found in the
    DS RR (see "trust anchor"). Second, the resolver may use an
    authenticated public key to verify a DS RR and the DNSKEY RR to
    which the DS RR refers. Third, the resolver may be able to
    determine that a new public key has been signed by the private key
    corresponding to another public key that the resolver has
    verified. Note that the resolver must always be guided by local
    policy when deciding whether to authenticate a new public key,
    even if the local policy is simply to authenticate any new public
    key for which the resolver is able verify the signature.

Authoritative RRset: Within the context of a particular zone, an
    RRset is "authoritative" if and only if the owner name of the
    RRset lies within the subset of the name space that is at or below
    the zone apex and at or above the cuts that separate the zone from
    its children, if any. All RRsets at the zone apex are
    authoritative, except for certain RRsets at this domain name that,
    if present, belong to this zone's parent. These RRset could
    include a DS RRset, the NSEC RRset referencing this DS RRset (the
    "parental NSEC"), and RRSIG RRs associated with these RRsets, all
    of which are authoritative in the parent zone. Similarly, if this
    zone contains any delegation points, only the parental NSEC RRset,
    DS RRsets, and any RRSIG RRs associated with these RRsets are
    authoritative for this zone.

Delegation Point: Term used to describe the name at the parental side
    of a zone cut. That is, the delegation point for "foo.example"
    would be the foo.example node in the "example" zone (as opposed to
    the zone apex of the "foo.example" zone). See also zone apex.

Island of Security: Term used to describe a signed, delegated zone
    that does not have an authentication chain from its delegating
    parent. That is, there is no DS RR containing a hash of a DNSKEY
    RR for the island in its delegating parent zone (see [RFC4034]).
    An island of security is served by security-aware name servers and
    may provide authentication chains to any delegated child zones.
    Responses from an island of security or its descendents can only
    be authenticated if its authentication keys can be authenticated
    by some trusted means out of band from the DNS protocol.

Key Signing Key (KSK): An authentication key that corresponds to a
    private key used to sign one or more other authentication keys for
    a given zone. Typically, the private key corresponding to a key
    signing key will sign a zone signing key, which in turn has a
    corresponding private key that will sign other zone data. Local
    policy may require that the zone signing key be changed
    frequently, while the key signing key may have a longer validity
    period in order to provide a more stable secure entry point into
    the zone. Designating an authentication key as a key signing key
    is purely an operational issue: DNSSEC validation does not
    distinguish between key signing keys and other DNSSEC
    authentication keys, and it is possible to use a single key as
    both a key signing key and a zone signing key. Key signing keys
    are discussed in more detail in [RFC3757]. Also see zone signing
    key.

   Non-Validating Security-Aware Stub Resolver: A security-aware stub
      resolver that trusts one or more security-aware recursive name
      servers to perform most of the tasks discussed in this document
      set on its behalf. In particular, a non-validating security-aware
      stub resolver is an entity that sends DNS queries, receives DNS
      responses, and is capable of establishing an appropriately secured
      channel to a security-aware recursive name server that will
      provide these services on behalf of the security-aware stub
      resolver. See also security-aware stub resolver, validating
      security-aware stub resolver.

   Non-Validating Stub Resolver: A less tedious term for a
      non-validating security-aware stub resolver.

   Security-Aware Name Server: An entity acting in the role of a name
      server (defined in section 2.4 of [RFC1034]) that understands the
      DNS security extensions defined in this document set. In
      particular, a security-aware name server is an entity that
      receives DNS queries, sends DNS responses, supports the EDNS0
      ([RFC2671]) message size extension and the DO bit ([RFC3225]), and
      supports the RR types and message header bits defined in this
      document set.

   Security-Aware Recursive Name Server: An entity that acts in both the
      security-aware name server and security-aware resolver roles. A
      more cumbersome but equivalent phrase would be "a security-aware
      name server that offers recursive service".

   Security-Aware Resolver: An entity acting in the role of a resolver
      (defined in section 2.4 of [RFC1034]) that understands the DNS
      security extensions defined in this document set. In particular,
      a security-aware resolver is an entity that sends DNS queries,
      receives DNS responses, supports the EDNS0 ([RFC2671]) message
      size extension and the DO bit ([RFC3225]), and is capable of using
      the RR types and message header bits defined in this document set
      to provide DNSSEC services.

   Security-Aware Stub Resolver: An entity acting in the role of a stub
      resolver (defined in section 5.3.1 of [RFC1034]) that has enough
      of an understanding the DNS security extensions defined in this
      document set to provide additional services not available from a
      security-oblivious stub resolver. Security-aware stub resolvers
      may be either "validating" or "non-validating", depending on
      whether the stub resolver attempts to verify DNSSEC signatures on
      its own or trusts a friendly security-aware name server to do so.
      See also validating stub resolver, non-validating stub resolver.

   Security-Oblivious <anything>: An <anything> that is not
      "security-aware".

   Signed Zone: A zone whose RRsets are signed and that contains
      properly constructed DNSKEY, Resource Record Signature (RRSIG),
      Next Secure (NSEC), and (optionally) DS records.

   Trust Anchor: A configured DNSKEY RR or DS RR hash of a DNSKEY RR. A
      validating security-aware resolver uses this public key or hash as
      a starting point for building the authentication chain to a signed
      DNS response. In general, a validating resolver will have to
      obtain the initial values of its trust anchors via some secure or
      trusted means outside the DNS protocol. Presence of a trust
      anchor also implies that the resolver should expect the zone to
      which the trust anchor points to be signed.

   Unsigned Zone: A zone that is not signed.

   Validating Security-Aware Stub Resolver: A security-aware resolver
      that sends queries in recursive mode but that performs signature
      validation on its own rather than just blindly trusting an
      upstream security-aware recursive name server. See also
      security-aware stub resolver, non-validating security-aware stub
      resolver.

   Validating Stub Resolver: A less tedious term for a validating
      security-aware stub resolver.

   Zone Apex: Term used to describe the name at the child's side of a
      zone cut. See also delegation point.

   Zone Signing Key (ZSK): An authentication key that corresponds to a
      private key used to sign a zone. Typically, a zone signing key
      will be part of the same DNSKEY RRset as the key signing key whose
      corresponding private key signs this DNSKEY RRset, but the zone
      signing key is used for a slightly different purpose and may
      differ from the key signing key in other ways, such as validity
      lifetime. Designating an authentication key as a zone signing key
      is purely an operational issue; DNSSEC validation does not
      distinguish between zone signing keys and other DNSSEC
      authentication keys, and it is possible to use a single key as
      both a key signing key and a zone signing key. See also key
      signing key.

3. Services Provided by DNS Security

   The Domain Name System (DNS) security extensions provide origin
   authentication and integrity assurance services for DNS data,
   including mechanisms for authenticated denial of existence of DNS
   data. These mechanisms are described below.

   These mechanisms require changes to the DNS protocol. DNSSEC adds
   four new resource record types: Resource Record Signature (RRSIG),
   DNS Public Key (DNSKEY), Delegation Signer (DS), and Next Secure
   (NSEC). It also adds two new message header bits: Checking Disabled
   (CD) and Authenticated Data (AD). In order to support the larger DNS
   message sizes that result from adding the DNSSEC RRs, DNSSEC also
   requires EDNS0 support ([RFC2671]). Finally, DNSSEC requires support
   for the DNSSEC OK (DO) EDNS header bit ([RFC3225]) so that a
   security-aware resolver can indicate in its queries that it wishes to
   receive DNSSEC RRs in response messages.

   These services protect against most of the threats to the Domain Name
   System described in [RFC3833]. Please see Section 12 for a
   discussion of the limitations of these extensions.

3.1. Data Origin Authentication and Data Integrity

   DNSSEC provides authentication by associating cryptographically
   generated digital signatures with DNS RRsets. These digital
   signatures are stored in a new resource record, the RRSIG record.
   Typically, there will be a single private key that signs a zone's
   data, but multiple keys are possible. For example, there may be keys
   for each of several different digital signature algorithms. If a
   security-aware resolver reliably learns a zone's public key, it can
   authenticate that zone's signed data. An important DNSSEC concept is
   that the key that signs a zone's data is associated with the zone
   itself and not with the zone's authoritative name servers. (Public
   keys for DNS transaction authentication mechanisms may also appear in
   zones, as described in [RFC2931], but DNSSEC itself is concerned with
   object security of DNS data, not channel security of DNS
   transactions. The keys associated with transaction security may be
   stored in different RR types. See [RFC3755] for details.)

   A security-aware resolver can learn a zone's public key either by
   having a trust anchor configured into the resolver or by normal DNS
   resolution. To allow the latter, public keys are stored in a new
   type of resource record, the DNSKEY RR. Note that the private keys
   used to sign zone data must be kept secure and should be stored
   offline when practical. To discover a public key reliably via DNS
   resolution, the target key itself has to be signed by either a
   configured authentication key or another key that has been

authenticated previously. Security-aware resolvers authenticate zone
information by forming an authentication chain from a newly learned
public key back to a previously known authentication public key,
which in turn either has been configured into the resolver or must
have been learned and verified previously. Therefore, the resolver
must be configured with at least one trust anchor.

If the configured trust anchor is a zone signing key, then it will
authenticate the associated zone; if the configured key is a key
signing key, it will authenticate a zone signing key. If the
configured trust anchor is the hash of a key rather than the key
itself, the resolver may have to obtain the key via a DNS query. To
help security-aware resolvers establish this authentication chain,
security-aware name servers attempt to send the signature(s) needed
to authenticate a zone's public key(s) in the DNS reply message along
with the public key itself, provided that there is space available in
the message.

The Delegation Signer (DS) RR type simplifies some of the
administrative tasks involved in signing delegations across
organizational boundaries. The DS RRset resides at a delegation
point in a parent zone and indicates the public key(s) corresponding
to the private key(s) used to self-sign the DNSKEY RRset at the
delegated child zone's apex. The administrator of the child zone, in
turn, uses the private key(s) corresponding to one or more of the
public keys in this DNSKEY RRset to sign the child zone's data. The
typical authentication chain is therefore
DNSKEY->[DS->DNSKEY]*->RRset, where "*" denotes zero or more
DS->DNSKEY subchains. DNSSEC permits more complex authentication
chains, such as additional layers of DNSKEY RRs signing other DNSKEY
RRs within a zone.

A security-aware resolver normally constructs this authentication
chain from the root of the DNS hierarchy down to the leaf zones based
on configured knowledge of the public key for the root. Local
policy, however, may also allow a security-aware resolver to use one
or more configured public keys (or hashes of public keys) other than
the root public key, may not provide configured knowledge of the root
public key, or may prevent the resolver from using particular public
keys for arbitrary reasons, even if those public keys are properly
signed with verifiable signatures. DNSSEC provides mechanisms by
which a security-aware resolver can determine whether an RRset's
signature is "valid" within the meaning of DNSSEC. In the final
analysis, however, authenticating both DNS keys and data is a matter
of local policy, which may extend or even override the protocol
extensions defined in this document set. See Section 5 for further
discussion.

3.2. Authenticating Name and Type Non-Existence

   The security mechanism described in Section 3.1 only provides a way
   to sign existing RRsets in a zone. The problem of providing negative
   responses with the same level of authentication and integrity
   requires the use of another new resource record type, the NSEC
   record. The NSEC record allows a security-aware resolver to
   authenticate a negative reply for either name or type non-existence
   with the same mechanisms used to authenticate other DNS replies. Use
   of NSEC records requires a canonical representation and ordering for
   domain names in zones. Chains of NSEC records explicitly describe
   the gaps, or "empty space", between domain names in a zone and list
   the types of RRsets present at existing names. Each NSEC record is
   signed and authenticated using the mechanisms described in Section
   3.1.

4. Services Not Provided by DNS Security

   DNS was originally designed with the assumptions that the DNS will
   return the same answer to any given query regardless of who may have
   issued the query, and that all data in the DNS is thus visible.
   Accordingly, DNSSEC is not designed to provide confidentiality,
   access control lists, or other means of differentiating between
   inquirers.

   DNSSEC provides no protection against denial of service attacks.
   Security-aware resolvers and security-aware name servers are
   vulnerable to an additional class of denial of service attacks based
   on cryptographic operations. Please see Section 12 for details.

   The DNS security extensions provide data and origin authentication
   for DNS data. The mechanisms outlined above are not designed to
   protect operations such as zone transfers and dynamic update
   ([RFC2136], [RFC3007]). Message authentication schemes described in
   [RFC2845] and [RFC2931] address security operations that pertain to
   these transactions.

5. Scope of the DNSSEC Document Set and Last Hop Issues

   The specification in this document set defines the behavior for zone
   signers and security-aware name servers and resolvers in such a way
   that the validating entities can unambiguously determine the state of
   the data.

   A validating resolver can determine the following 4 states:

   Secure: The validating resolver has a trust anchor, has a chain of
      trust, and is able to verify all the signatures in the response.

   Insecure: The validating resolver has a trust anchor, a chain of
      trust, and, at some delegation point, signed proof of the
      non-existence of a DS record. This indicates that subsequent
      branches in the tree are provably insecure. A validating resolver
      may have a local policy to mark parts of the domain space as
      insecure.

   Bogus: The validating resolver has a trust anchor and a secure
      delegation indicating that subsidiary data is signed, but the
      response fails to validate for some reason: missing signatures,
      expired signatures, signatures with unsupported algorithms, data
      missing that the relevant NSEC RR says should be present, and so
      forth.

   Indeterminate: There is no trust anchor that would indicate that a
      specific portion of the tree is secure. This is the default
      operation mode.

   This specification only defines how security-aware name servers can
   signal non-validating stub resolvers that data was found to be bogus
   (using RCODE=2, "Server Failure"; see [RFC4035]).

   There is a mechanism for security-aware name servers to signal
   security-aware stub resolvers that data was found to be secure (using
   the AD bit; see [RFC4035]).

   This specification does not define a format for communicating why
   responses were found to be bogus or marked as insecure. The current
   signaling mechanism does not distinguish between indeterminate and
   insecure states.

   A method for signaling advanced error codes and policy between a
   security-aware stub resolver and security-aware recursive nameservers
   is a topic for future work, as is the interface between a security-
   aware resolver and the applications that use it. Note, however, that
   the lack of the specification of such communication does not prohibit
   deployment of signed zones or the deployment of security aware
   recursive name servers that prohibit propagation of bogus data to the
   applications.

6. Resolver Considerations

   A security-aware resolver has to be able to perform cryptographic
   functions necessary to verify digital signatures using at least the
   mandatory-to-implement algorithm(s). Security-aware resolvers must
   also be capable of forming an authentication chain from a newly
   learned zone back to an authentication key, as described above. This
   process might require additional queries to intermediate DNS zones to

obtain necessary DNSKEY, DS, and RRSIG records. A security-aware
resolver should be configured with at least one trust anchor as the
starting point from which it will attempt to establish authentication
chains.

If a security-aware resolver is separated from the relevant
authoritative name servers by a recursive name server or by any sort
of intermediary device that acts as a proxy for DNS, and if the
recursive name server or intermediary device is not security-aware,
the security-aware resolver may not be capable of operating in a
secure mode. For example, if a security-aware resolver's packets are
routed through a network address translation (NAT) device that
includes a DNS proxy that is not security-aware, the security-aware
resolver may find it difficult or impossible to obtain or validate
signed DNS data. The security-aware resolver may have a particularly
difficult time obtaining DS RRs in such a case, as DS RRs do not
follow the usual DNS rules for ownership of RRs at zone cuts. Note
that this problem is not specific to NATs: any security-oblivious DNS
software of any kind between the security-aware resolver and the
authoritative name servers will interfere with DNSSEC.

If a security-aware resolver must rely on an unsigned zone or a name
server that is not security aware, the resolver may not be able to
validate DNS responses and will need a local policy on whether to
accept unverified responses.

A security-aware resolver should take a signature's validation period
into consideration when determining the TTL of data in its cache, to
avoid caching signed data beyond the validity period of the
signature. However, it should also allow for the possibility that
the security-aware resolver's own clock is wrong. Thus, a
security-aware resolver that is part of a security-aware recursive
name server will have to pay careful attention to the DNSSEC
"checking disabled" (CD) bit ([RFC4034]). This is in order to avoid
blocking valid signatures from getting through to other
security-aware resolvers that are clients of this recursive name
server. See [RFC4035] for how a secure recursive server handles
queries with the CD bit set.

7. Stub Resolver Considerations

Although not strictly required to do so by the protocol, most DNS
queries originate from stub resolvers. Stub resolvers, by
definition, are minimal DNS resolvers that use recursive query mode
to offload most of the work of DNS resolution to a recursive name
server. Given the widespread use of stub resolvers, the DNSSEC

architecture has to take stub resolvers into account, but the
security features needed in a stub resolver differ in some respects
from those needed in a security-aware iterative resolver.

Even a security-oblivious stub resolver may benefit from DNSSEC if
the recursive name servers it uses are security-aware, but for the
stub resolver to place any real reliance on DNSSEC services, the stub
resolver must trust both the recursive name servers in question and
the communication channels between itself and those name servers.
The first of these issues is a local policy issue: in essence, a
security-oblivious stub resolver has no choice but to place itself at
the mercy of the recursive name servers that it uses, as it does not
perform DNSSEC validity checks on its own. The second issue requires
some kind of channel security mechanism; proper use of DNS
transaction authentication mechanisms such as SIG(0) ([RFC2931]) or
TSIG ([RFC2845]) would suffice, as would appropriate use of IPsec.
Particular implementations may have other choices available, such as
operating system specific interprocess communication mechanisms.
Confidentiality is not needed for this channel, but data integrity
and message authentication are.

A security-aware stub resolver that does trust both its recursive
name servers and its communication channel to them may choose to
examine the setting of the Authenticated Data (AD) bit in the message
header of the response messages it receives. The stub resolver can
use this flag bit as a hint to find out whether the recursive name
server was able to validate signatures for all of the data in the
Answer and Authority sections of the response.

There is one more step that a security-aware stub resolver can take
if, for whatever reason, it is not able to establish a useful trust
relationship with the recursive name servers that it uses: it can
perform its own signature validation by setting the Checking Disabled
(CD) bit in its query messages. A validating stub resolver is thus
able to treat the DNSSEC signatures as trust relationships between
the zone administrators and the stub resolver itself.

8. Zone Considerations

There are several differences between signed and unsigned zones. A
signed zone will contain additional security-related records (RRSIG,
DNSKEY, DS, and NSEC records). RRSIG and NSEC records may be
generated by a signing process prior to serving the zone. The RRSIG
records that accompany zone data have defined inception and
expiration times that establish a validity period for the signatures
and the zone data the signatures cover.

8.1. TTL Values vs. RRSIG Validity Period

   It is important to note the distinction between a RRset's TTL value
   and the signature validity period specified by the RRSIG RR covering
   that RRset. DNSSEC does not change the definition or function of the
   TTL value, which is intended to maintain database coherency in
   caches. A caching resolver purges RRsets from its cache no later
   than the end of the time period specified by the TTL fields of those
   RRsets, regardless of whether the resolver is security-aware.

   The inception and expiration fields in the RRSIG RR ([RFC4034]), on
   the other hand, specify the time period during which the signature
   can be used to validate the covered RRset. The signatures associated
   with signed zone data are only valid for the time period specified by
   these fields in the RRSIG RRs in question. TTL values cannot extend
   the validity period of signed RRsets in a resolver's cache, but the
   resolver may use the time remaining before expiration of the
   signature validity period of a signed RRset as an upper bound for the
   TTL of the signed RRset and its associated RRSIG RR in the resolver's
   cache.

8.2. New Temporal Dependency Issues for Zones

   Information in a signed zone has a temporal dependency that did not
   exist in the original DNS protocol. A signed zone requires regular
   maintenance to ensure that each RRset in the zone has a current valid
   RRSIG RR. The signature validity period of an RRSIG RR is an
   interval during which the signature for one particular signed RRset
   can be considered valid, and the signatures of different RRsets in a
   zone may expire at different times. Re-signing one or more RRsets in
   a zone will change one or more RRSIG RRs, which will in turn require
   incrementing the zone's SOA serial number to indicate that a zone
   change has occurred and re-signing the SOA RRset itself. Thus,
   re-signing any RRset in a zone may also trigger DNS NOTIFY messages
   and zone transfer operations.

9. Name Server Considerations

   A security-aware name server should include the appropriate DNSSEC
   records (RRSIG, DNSKEY, DS, and NSEC) in all responses to queries
   from resolvers that have signaled their willingness to receive such
   records via use of the DO bit in the EDNS header, subject to message
   size limitations. Because inclusion of these DNSSEC RRs could easily
   cause UDP message truncation and fallback to TCP, a security-aware
   name server must also support the EDNS "sender's UDP payload"
   mechanism.

If possible, the private half of each DNSSEC key pair should be kept
offline, but this will not be possible for a zone for which DNS
dynamic update has been enabled. In the dynamic update case, the
primary master server for the zone will have to re-sign the zone when
it is updated, so the private key corresponding to the zone signing
key will have to be kept online. This is an example of a situation
in which the ability to separate the zone's DNSKEY RRset into zone
signing key(s) and key signing key(s) may be useful, as the key
signing key(s) in such a case can still be kept offline and may have
a longer useful lifetime than the zone signing key(s).

By itself, DNSSEC is not enough to protect the integrity of an entire
zone during zone transfer operations, as even a signed zone contains
some unsigned, nonauthoritative data if the zone has any children.
Therefore, zone maintenance operations will require some additional
mechanisms (most likely some form of channel security, such as TSIG,
SIG(0), or IPsec).

10. DNS Security Document Family

The DNSSEC document set can be partitioned into several main groups,
under the larger umbrella of the DNS base protocol documents.

The "DNSSEC protocol document set" refers to the three documents that
form the core of the DNS security extensions:

1. DNS Security Introduction and Requirements (this document)

2. Resource Records for DNS Security Extensions [RFC4034]

3. Protocol Modifications for the DNS Security Extensions [RFC4035]

Additionally, any document that would add to or change the core DNS
Security extensions would fall into this category. This includes any
future work on the communication between security-aware stub
resolvers and upstream security-aware recursive name servers.

The "Digital Signature Algorithm Specification" document set refers
to the group of documents that describe how specific digital
signature algorithms should be implemented to fit the DNSSEC resource
record format. Each document in this set deals with a specific
digital signature algorithm. Please see the appendix on "DNSSEC
Algorithm and Digest Types" in [RFC4034] for a list of the algorithms
that were defined when this core specification was written.

The "Transaction Authentication Protocol" document set refers to the
group of documents that deal with DNS message authentication,
including secret key establishment and verification. Although not

strictly part of the DNSSEC specification as defined in this set of
documents, this group is noted because of its relationship to DNSSEC.

The final document set, "New Security Uses", refers to documents that
seek to use proposed DNS Security extensions for other security
related purposes. DNSSEC does not provide any direct security for
these new uses but may be used to support them. Documents that fall
in this category include those describing the use of DNS in the
storage and distribution of certificates ([RFC2538]).

11. IANA Considerations

This overview document introduces no new IANA considerations. Please
see [RFC4034] for a complete review of the IANA considerations
introduced by DNSSEC.

12. Security Considerations

This document introduces DNS security extensions and describes the
document set that contains the new security records and DNS protocol
modifications. The extensions provide data origin authentication and
data integrity using digital signatures over resource record sets.
This section discusses the limitations of these extensions.

In order for a security-aware resolver to validate a DNS response,
all zones along the path from the trusted starting point to the zone
containing the response zones must be signed, and all name servers
and resolvers involved in the resolution process must be
security-aware, as defined in this document set. A security-aware
resolver cannot verify responses originating from an unsigned zone,
from a zone not served by a security-aware name server, or for any
DNS data that the resolver is only able to obtain through a recursive
name server that is not security-aware. If there is a break in the
authentication chain such that a security-aware resolver cannot
obtain and validate the authentication keys it needs, then the
security-aware resolver cannot validate the affected DNS data.

This document briefly discusses other methods of adding security to a
DNS query, such as using a channel secured by IPsec or using a DNS
transaction authentication mechanism such as TSIG ([RFC2845]) or
SIG(0) ([RFC2931]), but transaction security is not part of DNSSEC
per se.

A non-validating security-aware stub resolver, by definition, does
not perform DNSSEC signature validation on its own and thus is
vulnerable both to attacks on (and by) the security-aware recursive
name servers that perform these checks on its behalf and to attacks
on its communication with those security-aware recursive name

servers. Non-validating security-aware stub resolvers should use
some form of channel security to defend against the latter threat.
The only known defense against the former threat would be for the
security-aware stub resolver to perform its own signature validation,
at which point, again by definition, it would no longer be a
non-validating security-aware stub resolver.

DNSSEC does not protect against denial of service attacks. DNSSEC
makes DNS vulnerable to a new class of denial of service attacks
based on cryptographic operations against security-aware resolvers
and security-aware name servers, as an attacker can attempt to use
DNSSEC mechanisms to consume a victim's resources. This class of
attacks takes at least two forms. An attacker may be able to consume
resources in a security-aware resolver's signature validation code by
tampering with RRSIG RRs in response messages or by constructing
needlessly complex signature chains. An attacker may also be able to
consume resources in a security-aware name server that supports DNS
dynamic update, by sending a stream of update messages that force the
security-aware name server to re-sign some RRsets in the zone more
frequently than would otherwise be necessary.

Due to a deliberate design choice, DNSSEC does not provide
confidentiality.

DNSSEC introduces the ability for a hostile party to enumerate all
the names in a zone by following the NSEC chain. NSEC RRs assert
which names do not exist in a zone by linking from existing name to
existing name along a canonical ordering of all the names within a
zone. Thus, an attacker can query these NSEC RRs in sequence to
obtain all the names in a zone. Although this is not an attack on
the DNS itself, it could allow an attacker to map network hosts or
other resources by enumerating the contents of a zone.

DNSSEC introduces significant additional complexity to the DNS and
thus introduces many new opportunities for implementation bugs and
misconfigured zones. In particular, enabling DNSSEC signature
validation in a resolver may cause entire legitimate zones to become
effectively unreachable due to DNSSEC configuration errors or bugs.

DNSSEC does not protect against tampering with unsigned zone data.
Non-authoritative data at zone cuts (glue and NS RRs in the parent
zone) are not signed. This does not pose a problem when validating
the authentication chain, but it does mean that the non-authoritative
data itself is vulnerable to tampering during zone transfer
operations. Thus, while DNSSEC can provide data origin
authentication and data integrity for RRsets, it cannot do so for
zones, and other mechanisms (such as TSIG, SIG(0), or IPsec) must be
used to protect zone transfer operations.

Please see [RFC4034] and [RFC4035] for additional security
considerations.

13. Acknowledgements

This document was created from the input and ideas of the members of
the DNS Extensions Working Group. Although explicitly listing
everyone who has contributed during the decade in which DNSSEC has
been under development would be impossible, the editors would
particularly like to thank the following people for their
contributions to and comments on this document set: Jaap Akkerhuis,
Mark Andrews, Derek Atkins, Roy Badami, Alan Barrett, Dan Bernstein,
David Blacka, Len Budney, Randy Bush, Francis Dupont, Donald
Eastlake, Robert Elz, Miek Gieben, Michael Graff, Olafur Gudmundsson,
Gilles Guette, Andreas Gustafsson, Jun-ichiro Itojun Hagino, Phillip
Hallam-Baker, Bob Halley, Ted Hardie, Walter Howard, Greg Hudson,
Christian Huitema, Johan Ihren, Stephen Jacob, Jelte Jansen, Simon
Josefsson, Andris Kalnozols, Peter Koch, Olaf Kolkman, Mark Kosters,
Suresh Krishnaswamy, Ben Laurie, David Lawrence, Ted Lemon, Ed Lewis,
Ted Lindgreen, Josh Littlefield, Rip Loomis, Bill Manning, Russ
Mundy, Thomas Narten, Mans Nilsson, Masataka Ohta, Mike Patton, Rob
Payne, Jim Reid, Michael Richardson, Erik Rozendaal, Marcos Sanz,
Pekka Savola, Jakob Schlyter, Mike StJohns, Paul Vixie, Sam Weiler,
Brian Wellington, and Suzanne Woolf.

No doubt the above list is incomplete. We apologize to anyone we
left out.

14. References

14.1. Normative 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.

[RFC2535] Eastlake 3rd, D., "Domain Name System Security
           Extensions", RFC 2535, March 1999.

[RFC2671] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC
           2671, August 1999.

[RFC3225] Conrad, D., "Indicating Resolver Support of DNSSEC", RFC
           3225, December 2001.

   [RFC3226] Gudmundsson, O., "DNSSEC and IPv6 A6 aware server/resolver
              message size requirements", RFC 3226, December 2001.

   [RFC3445] Massey, D. and S. Rose, "Limiting the Scope of the KEY
              Resource Record (RR)", RFC 3445, December 2002.

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

   [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S.
              Rose, "Protocol Modifications for the DNS Security
              Extensions", RFC 4035, March 2005.

14.2. Informative References

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

   [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS
              Specification", RFC 2181, July 1997.

   [RFC2308] Andrews, M., "Negative Caching of DNS Queries (DNS
              NCACHE)", RFC 2308, March 1998.

   [RFC2538] Eastlake 3rd, D. and O. Gudmundsson, "Storing Certificates
              in the Domain Name System (DNS)", RFC 2538, March 1999.

   [RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B.
              Wellington, "Secret Key Transaction Authentication for DNS
              (TSIG)", RFC 2845, May 2000.

   [RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures
              (SIG(0)s)", RFC 2931, September 2000.

   [RFC3007] Wellington, B., "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.

   [RFC3090] Lewis, E., "DNS Security Extension Clarification on Zone
              Status", RFC 3090, March 2001.

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

   [RFC3655] Wellington, B. and O. Gudmundsson, "Redefinition of DNS
              Authenticated Data (AD) bit", RFC 3655, November 2003.

   [RFC3658] Gudmundsson, O., "Delegation Signer (DS) Resource Record
              (RR)", RFC 3658, December 2003.

   [RFC3755] Weiler, S., "Legacy Resolver Compatibility for Delegation
              Signer (DS)", RFC 3755, May 2004.

   [RFC3757] Kolkman, O., Schlyter, J., and E. Lewis, "Domain Name
              System KEY (DNSKEY) Resource Record (RR) Secure Entry
              Point (SEP) Flag", RFC 3757, April 2004.

   [RFC3833] Atkins, D. and R. Austein, "Threat Analysis of the Domain
              Name System (DNS)", RFC 3833, August 2004.

   [RFC3845] Schlyter, J., "DNS Security (DNSSEC) NextSECure (NSEC)
              RDATA Format", RFC 3845, August 2004.

Authors' Addresses

   Roy Arends
   Telematica Instituut
   Brouwerijstraat 1
   7523 XC Enschede
   NL

   EMail: roy.arends@telin.nl

   Rob Austein
   Internet Systems Consortium
   950 Charter Street
   Redwood City, CA 94063
   USA

   EMail: sra@isc.org

   Matt Larson
   VeriSign, Inc.
   21345 Ridgetop Circle
   Dulles, VA 20166-6503
   USA

   EMail: mlarson@verisign.com

   Dan Massey
   Colorado State University
   Department of Computer Science
   Fort Collins, CO 80523-1873

   EMail: massey@cs.colostate.edu

   Scott Rose
   National Institute for Standards and Technology
   100 Bureau Drive
   Gaithersburg, MD 20899-8920
   USA

   EMail: scott.rose@nist.gov

Full Copyright Statement

Intellectual Property

Acknowledgement