Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: October 16, 2015

P. Hoffman
VPN Consortium
A. Sullivan
Dyn
K. Fujiwara
JPRS

April 14, 2015

DNS Terminology
draft-ietf-dnsop-dns-terminology-00

Abstract

The DNS is defined in literally dozens of different RFCs. The terminology used in by implementers and developers of DNS protocols, and by operators of DNS systems, has sometimes changed in the decades since the DNS was first defined. This document gives current definitions for many of the terms used in the DNS in a single document.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on October 16, 2015.

Table of Contents

1. Introduction 2
2. Names 3
3. DNS Header and Response Codes 4
4. Resource Records 5
5. DNS Servers 7
6. Zones 10
7. Registration Model 13
8. General DNSSEC 14
9. DNSSEC States 15
10. IANA Considerations 17
11. Security Considerations 17
12. Acknowledgements 17
13. References 18
13.1. Normative References 18
13.2. Informative References 19
Authors' Addresses 20

1. Introduction

The domain name system (DNS) is a simple query-response protocol whose messages in both directions have the same format. The protocol and message format are defined in [RFC1034] and [RFC1035]. These RFCs defined some terms, but later documents defined others. Some of the terms from RFCs 1034 and 1035 now have somewhat different meanings than they did in 1987.

This document collects a wide variety of DNS-related terms. Some of them have been precisely defined in earlier RFCs, some have been loosely defined in earlier RFCs, and some are not defined in any earlier RFC at all.

The definitions here are believed to be the consensus definition of the DNS community, both protocol developers and operators. Some of the definitions differ from earlier RFCs, and those differences are noted. The terms are organized loosely by topic. Some definitions are for new terms for things that are commonly talked about in the DNS community but that never had terms defined for them.

In this document, where the consensus definition is the same as the one in an RFC, that RFC is quoted. Where the consensus definition

has changed somewhat, the RFC is mentioned but the new stand-alone definition is given.

Other organizations sometimes define DNS-related terms their own way. For example, the W3C defines "domain" at https://specs.webplatform.org/url/webspecs/develop/.

Note that there is no single consistent definition of "the DNS". It can be considered to be some combination of the following: a commonly-used naming scheme for objects on the Internet; a database representing the names and certain properties of these objects; an architecture providing distributed maintenance, resilience, and loose coherency for this database; and a simple query-response protocol (as mentioned in the current draft) implementing this architecture.

Capitalization in DNS terms is often inconsistent between RFCs and between DNS practitioners. The capitalization used in this document is a best guess at current practices, and is not meant to indicate that other capitalization styles are wrong or archaic.

2. Names

Domain name -- Section 3.1 of [RFC1034] talks of "the domain name space" as a tree structure. "Each node has a label, which is zero to 63 octets in length. ... The domain name of a node is the list of the labels on the path from the node to the root of the tree. ... To simplify implementations, the total number of octets that represent a domain name (i.e., the sum of all label octets and label lengths) is limited to 255."

Fully-qualified domain name (FQDN) -- This is often just a clear way of saying the same thing as "domain name of a node", as outlined above. However, the term is ambiguous. Strictly speaking, a fully-qualified name would include every label, including the final, zero-length label of the root zone: such a name would be written "www.example.net." (note the terminating dot). But because every name eventually shares the common root, names are often written relative to the root (such as "www.example.net") and are still called "fully qualified".
This term first appeared in [RFC1206].

Host name -- This term and its equivalent, "hostname", have been widely used but are not defined in [RFC1034], [RFC1035], [RFC1123], or [RFC2181]. The DNS was originally deployed into the Host Tables environment as outlined in [RFC0952], and it is likely that the term followed informally from the definition there. Over time, the definition seems to have shifted. "Host name" is often meant to be a domain name that follows the rules in Section 3.5 of [RFC1034], the

"preferred name syntax". Note that any label in any domain name can contain any octet value; hostnames are generally considered to be domain names where every label follows the rules in the "preferred name syntax", with the amendment that labels can start with ASCII digits (this amendment comes from Section 2.1 of [RFC1123]).

People also sometimes use the term hostname to refer to just the first label of an FQDN. In addition, people sometimes use this term to describe any name that refers to a machine, and those might include labels that do not conform to the "preferred name syntax".

TLD -- A Top-Level Domain, meaning a zone that is one layer below the root, such as .com or .jp. There is nothing special, from the point of view of the DNS, about TLDs. Most of them are also delegation-centric zones, and there are significant policy issues around their operation.

ccTLD -- A TLD that is allocated to a country. Historically, these were two-letter TLDs, and were allocated to countries using the two-letter code from the ISO 3166-1 alpha-2 standard [ISO3166]. In recent years, there have been allocations of TLDs that conform to IDNA2008 ([RFC5890], [RFC5891], [RFC5892], [RFC5893], and [RFC5894]); these are still treated as ccTLDs for policy purposes.

gTLD -- A "generic" TLD is a TLD that is not a ccTLD, and is not one of the small number of historical TLDs such as .int and .arpa. There is no precise definition for which TLDs that are not ccTLDs are gTLDs.

Public suffix -- A domain under which subdomains can be registered, and on which HTTP cookies ([RFC6265]) should not be set. For example, at the time this document is published, .com.au is considered a public suffix, but .au is not. Note that this term is controversial in the DNS community for many reasons, and may be significantly changed in the future. One example of the difficulty of calling a domain a public suffix is that designation can change over time as the registration policy for the zone changes, such as the case of the .uk zone around the time this document is published.

3. DNS Header and Response Codes

The header of a DNS message is first 12 octets. Many of the fields and flags in the header diagram in section 4.1.1 of [RFC1035] are referred to by their names in that diagram. For example, the response codes are called "RCODEs", and the authoritative answer bit is often called "the AA flag" or "the AA bit".

Some of response codes that are defined in [RFC1035] have gotten their own shorthand names. Some common response code names that appear without reference to the numeric value are "FORMERR", "SERVFAIL", and "NXDOMAIN". All of the RCODEs are listed at http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml, although that site uses mixed-case capitalization, while most documents use all-caps.

NODATA - This is not an actual response code, but is a particular type of response from a server that indicates that the queried domain name exists for the given class, but the resource record type being queried for does not exist. A NODATA response is a combination of an RCODE of 0 (NOERROR) and an Answer section that is empty. In addition, NODATA responses from authoritative servers have the authoritative answer (AA bit) set to 1 and include an SOA record. Section 1 of [RFC2308] defines NODATA as "a pseudo RCODE which indicates that the name is valid, for the given class, but are no records of the given type". The term "NXRRSET" is becoming more common as a synonym for NODATA.

Negative response -- A response whose RCODE indicates that a particular RRset does not exist in the DNS or a failure of a nameserver. Sections 2 and 7 of [RFC2308] describes the types of negative responses in detail.

4. Resource Records

RR -- A short form for resource record. ([RFC1034], section 3.6.)

RRset -- A set of resource records with the same label, class and type, but with different data. (Definition from [RFC2181]) Also spelled RRSet in some documents. As a clarification, "same label" in this definition means "same owner name". In addition, [RFC2181] states that "the TTLs of all RRs in an RRSet must be the same".

EDNS -- Also commonly called "EDNS0", this is the extension mechanisms for DNS. The extension mechanism, defined in [RFC6891], allows DNS clients and servers to specify message sizes larger than the original 512 octet limit, to expand the response code space, and to potentially carry additional options that affect the handling of a DNS query.

OPT -- A pseudo-RR (sometimes called a meta-RR) that is used only to contain control information pertaining to the question-and-answer sequence of a specific transaction. (Definition from [RFC6891], section 6.1.1) It is used by EDNS.

Owner -- The domain name where a RR is found ([RFC1034], section 3.6). Often appears in the term "owner name".

SOA field names -- DNS documents, including the definitions here, often refer to the fields in the RDATA an SOA resource record by field name. Those fields are defined in Section 3.3.13 of [RFC1035]. The names (in the order they appear in the SOA RDATA) are MNAME, RNAME, SERIAL, REFRESH, RETRY, EXPIRE, and MINIMUM. Note that the meaning of MINIMUM field is updated in Section 4 of [RFC2308]; the new definition is that the MINIMUM field is only "the TTL to be used for negative responses".

TTL -- The maximum "time to live" of a resource record. A TTL value is an unsigned number, with a minimum value of 0, and a maximum value of 2147483647. That is, a maximum of 2^31 - 1. When transmitted, the TTL is encoded in the less significant 31 bits of the 32 bit TTL field, with the most significant, or sign, bit set to zero. (Quoted from [RFC2181], section 8) (Note that [RFC1035] erroneously stated that this is a signed integer; it is fixed in an erratum.)

The TTL "specifies the time interval that the resource record may be cached before the source of the information should again be consulted". (Quoted from [RFC1035], section 3.2.1) Also: "the time interval (in seconds) that the resource record may be cached before it should be discarded". (Quoted from [RFC1035], section 4.1.3). Despite being defined for a resource record, the TTL of every resource record in an RRset is required to be the same (RFC2181, section 5.2).

The reason that the TTL is the maximum time to live is that a cache operator might decide to shorten the time to live for operational purposes, such as if there is a policy to not allow TTL values over a certain number. Also, if a value is flushed from the cache when its value is still positive, the value effectively becomes zero. Some servers do not honor the TTL on an RRset from the authoritative servers, such as when when the authoritative data has a very short TTL.

There is also the concept of a "default TTL" for a zone, which can be a configuration parameter in the server software. This is often expressed by a default for the entire server, and a default for a zone using the $TTL directive in a zone file. The $TTL directive was added to the master file format by [RFC2308].

5. DNS Servers

This section defines the terms used for the systems that act as DNS clients, DNS servers, or both. Some terms about servers describe servers that do and do not use DNSSEC; see Section 8 for those definitions.

[[There is a request to "first describe the iterative and recursive resolution processes, and mention the expected values of the RD,RA,AA bits. Then you can describe the distinctions between recursive and iterative clients, and between recursive and authoritative servers, in terms of the roles they play in the different resolution processes." This would require the section to be quite different than the other sections in the document.]]

Resolver -- A program that extracts information from name servers in response to client requests. (Quoted from [RFC1034], section 2.4) It is a program that interfaces user programs to domain name servers. The resolver is located on the same machine as the program that requests the resolver's services. (Quoted from [RFC1034], section 5.1) A resolver performs queries for a name, type, and class, and receives answers. The logical function is called "resolution". In practice, the term is usually referring to some specific type of resolver (some of which are defined below), and understanding the use of the term depends on understanding the context.

Stub resolver -- A resolver that cannot perform all resolution itself. Stub resolvers generally depend on a recursive resolver to undertake the actual resolution function. Stub resolvers are discussed but never fully defined in Section 5.3.1 of [RFC1034]. They are fully defined in Section 6.1.3.1 of [RFC1123].

Iterative mode -- A resolution mode of a server that receives DNS queries and responds with a referral to another server. Section 2.3 of [RFC1034] describes this as "The server refers the client to another server and lets the client pursue the query". A resolver that works in iterative mode is sometimes called an "iterative resolver".

Recursive mode -- A resolution mode of a server that receives DNS queries and either responds to those queries from a local cache or sends queries to other servers in order to get the final answers to the original queries. Section 2.3 of [RFC1034] describes this as "The first server pursues the query for the client at another server". A server operating in recursive mode may be thought of as having a name server side (which is what answers the query) and a resolver side (which performs the resolution function). Systems operating in this mode are commonly called "recursive servers".

Sometimes they are called "recursive resolvers". While strictly the difference between these is that one of them sends queries to another recursive server and the other does not, in practice it is not possible to know in advance whether the server that one is querying will also perform recursion; both terms can be observed in use interchangeably. Resolvers acting in recursive mode are also sometimes called "caching servers".

Full-service resolver -- Section 6.1.3.1 of [RFC1123] defines this term to mean a resolver that acts in recursive mode with a cache, as well as other requirements.

Priming -- The mechanism used by a resolver to determine where to send queries before there is anything in the resolver's cache. Priming is most often done from a configuration setting that contains a list of authoritative servers for the DNS root zone.

Negative caching -- The storage of knowledge that something does not exist, cannot give an answer, or does not give an answer. (Quoted from Section 1 of [RFC2308])

Authoritative server -- A system that responds to DNS queries with information about zones for which it has been configured to answer with the AA flag in the response header set to 1. It is a server that has authority over one or more DNS zones. Note that it is possible for an authoritative server to respond to a query without the parent zone delegating authority to that server. Authoritative servers also provide "referrals", usually to child zones delegated from them; these referrals have the AA bit set to 0 and come with referral data in the Additional section.

Primary servers and secondary servers -- These are synonyms for "master server" and "slave server", which were the terms used in the early DNS RFCs, and defined below. The current common usage has shifted to "primary" and "secondary".

Slave server -- An authoritative server which uses zone transfer to retrieve the zone. (Quoted from [RFC1996], section 2.1) [RFC2182] describes slave servers in detail.

Master -- Any authoritative server configured to be the source of zone transfer for one or more slave servers. (Quoted from [RFC1996], section 2.1) [RFC2136] defines "master" as "an authoritative server configured to be the source of AXFR or IXFR data for one or more slave servers".

Primary master -- The primary master is named in the zone's SOA MNAME field and optionally by an NS resource record. (Quoted from

[RFC1996], section 2.1) [RFC2136] defines "primary master" as "Master server at the root of the AXFR/IXFR dependency graph. The primary master is named in the zone's SOA MNAME field and optionally by an NS RR. There is by definition only one primary master server per zone."

Stealth server -- This is the same as a slave server except that it is not listed in an NS resource record for the zone. (Quoted from [RFC1996], section 2.1) A stealth server is often actually a master for zone transfers, and in that case is called a "hidden master".

Zone transfer -- The act of a client requesting a copy of a zone and an authoritative server sending the needed information. There are two common standard ways to do zone transfers: the AXFR ("Authoritative Transfer") mechanism to copy the full zone, and the IXFR ("Incremental Transfer") mechanism to copy only parts of the zone that have changed. Many systems use non-standard methods for zone transfer outside the DNS protocol.

Forwarder -- A system that receives a DNS query, possibly changes the query, sends on the resulting query (usually to a recursive resolver), receives the response, possibly changes the response, and sends the resulting response to the source of the query (usually a stub resolver). Section 1 of [RFC2308] describes a forwarder as "a nameserver used to resolve queries instead of directly using the authoritative nameserver chain". [RFC2308] further says "The forwarder typically either has better access to the internet, or maintains a bigger cache which may be shared amongst many resolvers."

[RFC5625] does not give a specific definition for DNS forwarder, but describes in detail what features they need to support. The protocol interfaces for DNS forwarders are exactly the same as those for recursive resolvers (for interactions with DNS stubs) and as those for stub resolvers (for interactions with recursive resolvers). Forwarders are also sometimes called "DNS forwarders". They are sometimes also called "DNS proxies", but that term has not yet been defined (even in [RFC5625]).

Full resolver -- This term is used in [RFC1035], but it is not defined there. RFC 1123 defines a "full-service resolver" that may or may not be what was intended by "full resolver" in [RFC1035]. In the vernacular, a full-service resolver is usually one that would be suitable for use by a stub resolver.

Consensual policy-implementing resolver -- A resolver that changes some answers it returns based on policy criteria, such as to prevent access to malware sites. These policy criteria are agreed to by systems that query this resolver through some out of band mechanism

(such as finding out about the resolver from a web site and reading the policy).

Non-consensual policy-implementing resolver -- A resolver that is not a consensual policy-implementing resolver that changes the answers it returns. The difference between this and a consensual policy-implementing resolver is that users of this resolver are not expected to know that there is a policy to change the answers it returns.

Open resolver -- A full resolver that accepts and processes queries from any (or nearly any) stub resolver. This is sometimes also called a "public resolver".

Open forwarder -- A DNS forwarder that accepts and forwards queries from any (or nearly any) stub resolver to a full resolver.

View -- A configuration for a DNS server that allows it to provide different answers depending on attributes of the query. Typically, views differ by the source IP address of a query, but can also be based on the destination IP address, the type of query (such as AXFR), and so on. Views are often used to provide more names or different addresses to queries from "inside" a protected network than to those "outside" that network. Views are not a standardized part of the DNS, but they are widely implemented in server software.

Passive DNS -- A mechanism to collect large amounts of DNS data by storing queries and responses from recursive servers. Passive DNS databases can be used to answer historical questions about DNS zones such as which records were available for them at what times in the past. Passive DNS databases allow searching of the stored records on keys other than just the name, such as "find all names which have A records of a particular value".

Child-centric resolver -- A DNS resolver that, instead of serving the NS RRset and glue records that it obtained from the parent of a zone, serves data from the authoritative servers for that zone. The term "child-centric" is meant as the opposite of "parent-centric", which means a resolver that simply serves the NS RRset and glue records for a zone that it obtained from the zone's parent, without checking the authoritative servers for that zone.

6. Zones

This section defines terms that are used when discussing zones that are being served or retrieved.

Zone -- A unit of organization of authoritative data. Zones can be automatically distributed to the name servers which provide redundant

service for the data in a zone. (Quoted from [RFC1034], section 2.4).

Child -- The entity on record that has the delegation of the domain from the Parent. (Quoted from [RFC7344], section 1.1)

Parent -- The domain in which the Child is registered. (Quoted from [RFC7344], section 1.1) Earlier, "parent name server" was defined in [RFC0882] as "the name server that has authority over the place in the domain name space that will hold the new domain".

Origin -- 1. The domain name that appears at the top of a zone. 2. The domain name within which a given relative domain name appears in zone files. Generally seen in the context of "$ORIGIN", which is a control entry defined in [RFC1035], section 5.1, as part of the master file format. For example, if the $ORIGIN is set to "example.org.", then a master file line for "www" is in fact an entry for "www.example.org.".

Zone cut -- The delimitation point between two zones where the origin of one of the zones is the child of the other zone. (Section 6 of [RFC2181] uses this term extensively, although never actually defines it.) Section 4.2 of [RFC1034] uses "cuts" as "zone cut".

Apex -- The point in the tree at an owner of an SOA and corresponding authoritative NS RRset. This is also called the "zone apex". [RFC4033] defines it as "the name at the child's side of a zone cut". The "apex" can usefully be thought of as a data-theoretic description of a tree structure, and "origin" is the name of the same concept when it is implemented in zone files. The distinction is not always maintained in use, however, and one can find uses that conflict subtly with this definition.

Delegation -- The process by which a separate zone is created in the name space beneath the apex of a given domain. Delegation happens when an NS RRset is added in the parent zone for the child origin, and a corresponding zone apex is created at the child origin. Delegation inherently happens at a zone cut.

Referrals -- Data from the authority section of a non-authoritative answer. [RFC1035] section 2.1 defines "authoritative" data. However, referrals at zone cuts are not authoritative. Referrals may be a zone cut NS resource records and their glue. NS records on the parent side of a zone cut are an authoritative delegation, but are normally not treated as authoritative data by the client. In general, a referral is a way for a server to send an answer saying that the server does not know the answer, but knows where the query should be directed in order to get an answer. Historically, many

authoritative servers answered with a referral to the root zone when queried for a name for which they were not authoritative, but this practice has declined.

Glue records -- Resource records which are not part of the authoritative data [for a zone], and are address resource records for the servers [in a subzone]. These RRs are only necessary if the name server's name is "below" the cut, and are only used as part of a referral response. (Definition from [RFC1034], section 4.2.1)

A later definition is that glue "includes any record in a zone file that is not properly part of that zone, including nameserver records of delegated sub-zones (NS records), address records that accompany those NS records (A, AAAA, etc), and any other stray data that might appear". (Definition from [RFC2181], section 5.4.1) This definition in [RFC2181] is wider than the one from [RFC1034], and bases the definition on the contents of a zone file. Some implementers might only be thinking about the earlier definition when they see rules about glue records.

In-bailiwick - 1. An adjective to describe a name server the name of which is either subordinate to or (rarely) the same as the zone origin. In-bailiwick name servers require glue in their parent zone. 2. Data for which the server is either authoritative, or else authoritative for an ancestor of the owner name. This sense of the term normally is used when discussing the relevancy of glue records in a response. For example, the server for the parent zone example.com might reply with glue records for ns.child.example.com. Because the child.example.com zone is a descendant of the example.com zone, both glue records are in-bailiwick.

Out-of-bailiwick - The antonym of in-bailiwick.

Authoritative data -- All of the RRs attached to all of the nodes from the top node of the zone down to leaf nodes or nodes above cuts around the bottom edge of the zone. (Quoted from Section 4.2.1 of [RFC1034]) It is noted that this definition might inadvertently also include any NS records that appear in the zone, even those that might not truly be authoritative because there are identical NS RRs below the zone cut. This reveals the ambiguity in the notion of authoritative data, because the parent-size NS records authoritatively indicate the delegation, even though they are not themselves authoritative data.

Root zone -- The zone whose origin is the zero-length label. Also sometimes called "the DNS root".

Empty non-terminal -- A domain name that has no RRsets, but has descendants that have RRsets. A typical example is in SRV records: in the name "_sip._tcp.example.com", it is likely that "_tcp.example.com" has no RRsets, but that "_sip._tcp.example.com" has (at least) an SRV RRset.

Delegation-centric zone -- A zone which consists mostly of delegations to child zones. This term is used in contrast to a zone which might have some delegations to child zones, but also has many data resource records for the zone itself and/or for child zones.

Wildcard -- [RFC1034] defined "wildcard", but in a way that turned out to be confusing to implementers. For an extended discussion of wildcards, including clearer definitions, see [RFC4592].

Occluded name -- The addition of a delegation point via dynamic update will render all subordinate domain names to be in a limbo, still part of the zone but not available to the lookup process. The addition of a DNAME resource record has the same impact. The subordinate names are said to be "occluded". (Quoted from [RFC5936], Section 3.5)

Fast flux DNS -- A mechanism where a large number of hosts rapidly update the address records of a zone, often to deliver malware. Because the addresses change so rapidly, it is difficult to definitively find all the hosts.

7. Registration Model

Registry -- The administrative operation of a zone that allows registration of names within that zone.

Registrant -- An individual or organization on whose behalf a name in a zone is registered by the registry. In many zones, the registry and the registrant may be the same entity, but in TLDs they often are not.

Registrar -- A service provider that acts as a go-between for registrants and registries. Not all registrations require a registrar, though it is common to have registrars be involved in registrations in TLDs.

EPP -- The Extensible Provisioning Protocol (EPP), which is commonly used for communication of registration information between registries and registrars. EPP is defined in [RFC5730].

8. General DNSSEC

Most DNSSEC terms are defined in [RFC4033], [RFC4034], and [RFC4035]. The terms that have caused confusion in the DNS community are highlighted here.

DNSSEC-aware and DNSSEC-unaware -- Section 2 of [RFC4033] defines many types of resolvers and validators. In specific, the terms "non-validating security-aware stub resolver", "non-validating stub resolver", "security-aware name server", "security-aware recursive name server", "security-aware resolver", "security-aware stub resolver", and "security-oblivious 'anything'" are all defined. (Note that the term "validating resolver", which is used in some places in those documents, is nevertheless not defined in that section.)

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. (Quoted from [RFC4033], section 2) It has been noted in other contexts that the zone itself is not really signed, but all the relevant RRsets in the zone are signed. It should also be noted that, since the publication of [RFC6840], NSEC records are no longer required for signed zones: a signed zone might include NSEC3 records instead.

Unsigned zone -- Section 2 of [RFC4033] defines this as "a zone that is not signed". Section 2 of [RFC4035] defines this as "A zone that does not include these records [properly constructed DNSKEY, Resource Record Signature (RRSIG), Next Secure (NSEC), and (optionally) DS records] according to the rules in this section". There is an important note at the end of Section 5.2 of [RFC4035] adding an additional situation when a zone is considered unsigned: "If the resolver does not support any of the algorithms listed in an authenticated DS RRset, then the resolver will not be able to verify the authentication path to the child zone. In this case, the resolver SHOULD treat the child zone as if it were unsigned."

NSEC -- "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." (Quoted from Section 3.2 of [RFC4033]) In short, an NSEC record provides authenticated denial of existence.

The NSEC resource record lists two separate things: the next owner name (in the canonical ordering of the zone) that contains authoritative data or a delegation point NS RRset, and the set of RR types present at the NSEC RR's owner name. (Quoted from Section 4 of 4034)

NSEC3 -- The NSEC3 resource record is quite different than the NSEC resource record. Like the NSEC record, the NSEC3 record also provides authenticated denial of existence; however, NSEC3 records mitigates against zone enumeration and support Opt-Out. NSEC3 resource records are defined in [RFC5155].

Opt-out -- The Opt-Out Flag indicates whether this NSEC3 RR may cover unsigned delegations. (Quoted from Section 3.1.2.1 of [RFC5155])

Zone enumeration -- The practice of discovering the full content of a zone via successive queries. (Quoted from Section 1.3 of [RFC5155]) This is also sometimes call "zone walking". Zone enumeration is different from zone content guessing where the guesser uses a large dictionary of possible labels and sends successive queries for them, or matches the contents of NSEC3 records against such a dictionary.

DNSSEC Policy (DP) -- A statement that sets forth the security requirements and standards to be implemented for a DNSSEC-signed zone. (Quoted from [RFC6841], section 2)

DNSSEC Practice Statement (DPS) -- A practices disclosure document that may support and be a supplemental document to the DNSSEC Policy (if such exists), and it states how the management of a given zone implements procedures and controls at a high level. (Quoted from [RFC6841], section 2)

Key signing key (KSK) -- DNSSEC keys that only sign the apex DNSKEY RRset in a zone. (Quoted from [RFC6781], Section 3.1)

Zone signing key (ZSK) -- DNSSEC keys that can be used to sign all the RRsets in a zone that require signatures, other than the apex DNSKEY RRset. (Quoted from [RFC6781], Section 3.1) Note that the roles KSK and ZSK are not mutually exclusive: a single key can be both KSK and ZSK at the same time. This is sometimes called a "combined signing key" or CSK. It is operational practice, not protocol, that determines whether a particular key is a ZSK, a KSK, or a CSK.

9. DNSSEC States

A validating resolver can determine that a response is in one of four states: secure, insecure, bogus, or indeterminate. These states are defined in [RFC4033] and [RFC4035], although the two definitions differ a bit.

Section 5 of [RFC4033] says:

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.

Section 4.3 of [RFC4035] says:

A security-aware resolver must be able to distinguish between four cases:

Secure: An RRset for which the resolver is able to build a chain of signed DNSKEY and DS RRs from a trusted security anchor to the RRset. In this case, the RRset should be signed and is subject to signature validation, as described above.

Insecure: An RRset for which the resolver knows that it has no chain of signed DNSKEY and DS RRs from any trusted starting point to the RRset. This can occur when the target RRset lies in an unsigned zone or in a descendent of an unsigned zone. In this case, the RRset may or may not be signed, but the resolver will not be able to verify the signature.

Bogus: An RRset for which the resolver believes that it ought to be able to establish a chain of trust but for which it is unable to do so, either due to signatures that for some reason fail to validate or due to missing data that the relevant DNSSEC RRs indicate should be present. This case may indicate an attack but may also indicate a configuration error or some form of data corruption.

Indeterminate: An RRset for which the resolver is not able to determine whether the RRset should be signed, as the resolver is not able to obtain the necessary DNSSEC RRs. This can occur when the security-aware resolver is not able to contact security-aware name servers for the relevant zones.

10. IANA Considerations

This document has no effect on IANA registries.

11. Security Considerations

These definitions do not change any security considerations for the DNS.

12. Acknowledgements

The authors gratefully acknowledge all of the authors of DNS-related RFCs that proceed this one. Comments from Tony Finch, Stephane Bortzmeyer, Niall O'Reilly, Colm MacCarthaigh, Ray Bellis, John Kristoff, Robert Edmonds, Paul Wouters, Shumon Huque, Paul Ebersman, David Lawrence, Matthijs Mekking, Casey Deccio, and many others in the DNSOP Working Group have helped shape this document.

13. References

13.1. Normative References

[ISO3166] International Organization for Standardization (ISO), "Country Codes - ISO 3166", February 2015, <http://www.iso.org/iso/country_codes/country_codes>.

[RFC0882] Mockapetris, P., "Domain names: Concepts and facilities", RFC 882, November 1983.

[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.

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

[RFC1206] Malkin, G. and A. Marine, "FYI on Questions and Answers: Answers to commonly asked "new Internet user" questions", RFC 1206, February 1991.

[RFC1996] Vixie, P., "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", RFC 1996, August 1996.

[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.

[RFC2182] Elz, R., Bush, R., Bradner, S., and M. Patton, "Selection and Operation of Secondary DNS Servers", BCP 16, RFC 2182, July 1997.

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

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the 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.

[RFC4592] Lewis, E., "The Role of Wildcards in the Domain Name System", RFC 4592, July 2006.

[RFC5155] Laurie, B., Sisson, G., Arends, R., and D. Blacka, "DNS Security (DNSSEC) Hashed Authenticated Denial of Existence", RFC 5155, March 2008.

[RFC5730] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", STD 69, RFC 5730, August 2009.

[RFC5936] Lewis, E. and A. Hoenes, "DNS Zone Transfer Protocol (AXFR)", RFC 5936, June 2010.

[RFC6781] Kolkman, O., Mekking, W., and R. Gieben, "DNSSEC Operational Practices, Version 2", RFC 6781, December 2012.

[RFC6840] Weiler, S. and D. Blacka, "Clarifications and Implementation Notes for DNS Security (DNSSEC)", RFC 6840, February 2013.

[RFC6841] Ljunggren, F., Eklund Lowinder, AM., and T. Okubo, "A Framework for DNSSEC Policies and DNSSEC Practice Statements", RFC 6841, January 2013.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, April 2013.

[RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", RFC 7344, September 2014.

13.2. Informative References

[RFC0952] Harrenstien, K., Stahl, M., and E. Feinler, "DoD Internet host table specification", RFC 952, October 1985.

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, August 2009.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

[RFC5892] Faltstrom, P., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.

[RFC5893] Alvestrand, H. and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, August 2010.

[RFC5894] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, August 2010.

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, April 2011.

Authors' Addresses

Paul Hoffman
VPN Consortium
127 Segre Place
Santa Cruz, CA 95060
USA

Email: paul.hoffman@vpnc.org

Andrew Sullivan
Dyn
150 Dow St, Tower 2
Manchester, NH 1604
USA

Email: asullivan@dyn.com

Kazunori Fujiwara
Japan Registry Services Co., Ltd.
Chiyoda First Bldg. East 13F, 3-8-1 Nishi-Kanda
Chiyoda-ku, Tokyo 101-0065
Japan

Phone: +81 3 5215 8451
Email: fujiwara@jprs.co.jp