Network Working Group
Request for Comments: 3597
Category: Standards Track
A. Gustafsson
Nominum Inc.
September 2003

Handling of Unknown DNS Resource Record (RR) Types

Status of this Memo

Copyright Notice

Abstract

Extending the Domain Name System (DNS) with new Resource Record (RR) types currently requires changes to name server software. This document specifies the changes necessary to allow future DNS implementations to handle new RR types transparently.

1. Introduction

The DNS is designed to be extensible to support new services through the introduction of new resource record (RR) types. In practice, deploying a new RR type currently requires changes to the name server software not only at the authoritative DNS server that is providing the new information and the client making use of it, but also at all slave servers for the zone containing it, and in some cases also at caching name servers and forwarders used by the client.

Because the deployment of new server software is slow and expensive, the potential of the DNS in supporting new services has never been fully realized. This memo proposes changes to name servers and to procedures for defining new RR types aimed at simplifying the future deployment of new RR types.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119].

2. Definition

An "RR of unknown type" is an RR whose RDATA format is not known to the DNS implementation at hand, and whose type is not an assigned QTYPE or Meta-TYPE as specified in [RFC 2929] (section 3.1) nor within the range reserved in that section for assignment only to QTYPEs and Meta-TYPEs. Such an RR cannot be converted to a type-specific text format, compressed, or otherwise handled in a type-specific way.

In the case of a type whose RDATA format is class specific, an RR is considered to be of unknown type when the RDATA format for that combination of type and class is not known.

3. Transparency

To enable new RR types to be deployed without server changes, name servers and resolvers MUST handle RRs of unknown type transparently. That is, they must treat the RDATA section of such RRs as unstructured binary data, storing and transmitting it without change [RFC1123].

To ensure the correct operation of equality comparison (section 6) and of the DNSSEC canonical form (section 7) when an RR type is known to some but not all of the servers involved, servers MUST also exactly preserve the RDATA of RRs of known type, except for changes due to compression or decompression where allowed by section 4 of this memo. In particular, the character case of domain names that are not subject to compression MUST be preserved.

4. Domain Name Compression

RRs containing compression pointers in the RDATA part cannot be treated transparently, as the compression pointers are only meaningful within the context of a DNS message. Transparently copying the RDATA into a new DNS message would cause the compression pointers to point at the corresponding location in the new message, which now contains unrelated data. This would cause the compressed name to be corrupted.

To avoid such corruption, servers MUST NOT compress domain names embedded in the RDATA of types that are class-specific or not well-known. This requirement was stated in [RFC1123] without defining the term "well-known"; it is hereby specified that only the RR types defined in [RFC1035] are to be considered "well-known".

The specifications of a few existing RR types have explicitly allowed compression contrary to this specification: [RFC2163] specified that compression applies to the PX RR, and [RFC2535] allowed compression in SIG RRs and NXT RRs records. Since this specification disallows compression in these cases, it is an update to [RFC2163] (section 4) and [RFC2535] (sections 4.1.7 and 5.2).

Receiving servers MUST decompress domain names in RRs of well-known type, and SHOULD also decompress RRs of type RP, AFSDB, RT, SIG, PX, NXT, NAPTR, and SRV (although the current specification of the SRV RR in [RFC2782] prohibits compression, [RFC2052] mandated it, and some servers following that earlier specification are still in use).

Future specifications for new RR types that contain domain names within their RDATA MUST NOT allow the use of name compression for those names, and SHOULD explicitly state that the embedded domain names MUST NOT be compressed.

As noted in [RFC1123], the owner name of an RR is always eligible for compression.

5. Text Representation

In the "type" field of a master file line, an unknown RR type is represented by the word "TYPE" immediately followed by the decimal RR type number, with no intervening whitespace. In the "class" field, an unknown class is similarly represented as the word "CLASS" immediately followed by the decimal class number.

This convention allows types and classes to be distinguished from each other and from TTL values, allowing the "[<TTL>] [<class>] <type> <RDATA>" and "[<class>] [<TTL>] <type> <RDATA>" forms of [RFC1035] to both be unambiguously parsed.

The RDATA section of an RR of unknown type is represented as a sequence of white space separated words as follows:

The special token \# (a backslash immediately followed by a hash sign), which identifies the RDATA as having the generic encoding defined herein rather than a traditional type-specific encoding.

An unsigned decimal integer specifying the RDATA length in octets.

Zero or more words of hexadecimal data encoding the actual RDATA field, each containing an even number of hexadecimal digits.

If the RDATA is of zero length, the text representation contains only the \# token and the single zero representing the length.

An implementation MAY also choose to represent some RRs of known type using the above generic representations for the type, class and/or RDATA, which carries the benefit of making the resulting master file portable to servers where these types are unknown. Using the generic representation for the RDATA of an RR of known type can also be useful in the case of an RR type where the text format varies depending on a version, protocol, or similar field (or several) embedded in the RDATA when such a field has a value for which no text format is known, e.g., a LOC RR [RFC1876] with a VERSION other than 0.

Even though an RR of known type represented in the \# format is effectively treated as an unknown type for the purpose of parsing the RDATA text representation, all further processing by the server MUST treat it as a known type and take into account any applicable type-specific rules regarding compression, canonicalization, etc.

The following are examples of RRs represented in this manner, illustrating various combinations of generic and type-specific encodings for the different fields of the master file format:

```
a.example.   CLASS32     TYPE731         \# 6 abcd (
                                         ef 01 23 45 )
b.example.   HS          TYPE62347       \# 0
e.example.   IN          A               \# 4 0A000001
e.example.   CLASS1      TYPE1           10.0.0.2
```

6. Equality Comparison

Certain DNS protocols, notably Dynamic Update [RFC2136], require RRs to be compared for equality. Two RRs of the same unknown type are considered equal when their RDATA is bitwise equal. To ensure that the outcome of the comparison is identical whether the RR is known to the server or not, specifications for new RR types MUST NOT specify type-specific comparison rules.

This implies that embedded domain names, being included in the overall bitwise comparison, are compared in a case-sensitive manner.

As a result, when a new RR type contains one or more embedded domain names, it is possible to have multiple RRs owned by the same name that differ only in the character case of the embedded domain name(s). This is similar to the existing possibility of multiple TXT records differing only in character case, and not expected to cause any problems in practice.

7. DNSSEC Canonical Form and Ordering

DNSSEC defines a canonical form and ordering for RRs [RFC2535] (section 8.1). In that canonical form, domain names embedded in the RDATA are converted to lower case.

The downcasing is necessary to ensure the correctness of DNSSEC signatures when case distinctions in domain names are lost due to compression, but since it requires knowledge of the presence and position of embedded domain names, it cannot be applied to unknown types.

To ensure continued consistency of the canonical form of RR types where compression is allowed, and for continued interoperability with existing implementations that already implement the [RFC2535] canonical form and apply it to their known RR types, the canonical form remains unchanged for all RR types whose whose initial publication as an RFC was prior to the initial publication of this specification as an RFC (RFC 3597).

As a courtesy to implementors, it is hereby noted that the complete set of such previously published RR types that contain embedded domain names, and whose DNSSEC canonical form therefore involves downcasing according to the DNS rules for character comparisons, consists of the RR types NS, MD, MF, CNAME, SOA, MB, MG, MR, PTR, HINFO, MINFO, MX, HINFO, RP, AFSDB, RT, SIG, PX, NXT, NAPTR, KX, SRV, DNAME, and A6.

This document specifies that for all other RR types (whether treated as unknown types or treated as known types according to an RR type definition RFC more recent than RFC 3597), the canonical form is such that no downcasing of embedded domain names takes place, and otherwise identical to the canonical form specified in [RFC2535] section 8.1.

Note that the owner name is always set to lower case according to the DNS rules for character comparisons, regardless of the RR type.

The DNSSEC canonical RR ordering is as specified in [RFC2535] section 8.3, where the octet sequence is the canonical form as revised by this specification.

8. Additional Section Processing

Unknown RR types cause no additional section processing. Future RR type specifications MAY specify type-specific additional section processing rules, but any such processing MUST be optional as it can only be performed by servers for which the RR type in case is known.

9. IANA Considerations

This document does not require any IANA actions.

10. Security Considerations

This specification is not believed to cause any new security problems, nor to solve any existing ones.

11. Normative References

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

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

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

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

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

[RFC2163] Allocchio, C., "Using the Internet DNS to Distribute MIXER Conformant Global Address Mapping (MCGAM)", RFC 2163, January 1998.

[RFC2929] Eastlake, D., Brunner-Williams, E. and B. Manning, "Domain Name System (DNS) IANA Considerations", BCP 42, RFC 2929, September 2000.

12. Informative References

[RFC1876] Davis, C., Vixie, P., Goodwin, T. and I. Dickinson, "A Means for Expressing Location Information in the Domain Name System", RFC 1876, January 1996.

[RFC2052] Gulbrandsen, A. and P. Vixie, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2052, October 1996.

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

[RFC2782] Gulbrandsen, A., Vixie, P. and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

13. Intellectual Property Statement

14. Author's Address

Andreas Gustafsson
Nominum, Inc.
2385 Bay Rd
Redwood City, CA 94063
USA

Phone: +1 650 381 6004
EMail: gson@nominum.com

15. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.