Domain Name Operations
Internet-Draft
Intended status: Informational
Expires: May 24, 2019

L-J. Liman
Netnod
R. Sundblad
Royal Institute of Technology

November 20, 2018

A UTC Timestamp Option For EDNS
draft-liman-dns-utcstamp-00

Abstract

UTCSTAMP is an EDNS extension to allow a client to request from a server that it includes a timestamp in the response message, and for the server to provide it, if requested and deemed appropriate. This is primarily intended as a debugging tool.

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 https://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 May 24, 2019.

Copyright Notice

Table of Contents

1. Introduction

Network security based on encryption depends heavily on the requirement that all involved parties have a common understanding of the time of day. This is true also for the domain name system (DNS) and its transaction signature (TSIG) is no exception. If the time difference between the DNS server and the DNS client is too large, TSIG signatures will not validate. When debugging security-related issues with the DNS, knowing what a remote party believes to be the current time can be very helpful. This documents describes an option to Extended DNS (EDNS) [RFC6891] that allows a client to request that the server includes a timestamp in the response packet, and for the server to provide it, if requested and deemed appropriate.

This document is modeled after the NSID option, described in RFC 5001 [RFC5001].

2. Requirements Language

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 [RFC2119].

3. Protocol

This protocol uses an EDNS RFC 6891 [RFC6891] option to signal a resolver's desire for information identifying a server's understanding of the current date and time of day, and to hold the name server's response, if any.

3.1. General Behavior

The semantics of a UTCSTAMP request are not transitive. That means that any DNS server that receives a incoming query with a UTCSTAMP request, to which it intends to honour it, MUST report its own understanding of the current time, and not relay information obtained from a different host.

UTCSTAMP responses MUST NOT be cached.

3.2. Resolver Behavior

A resolver signals its desire for information about the server's understanding of the current time by sending an empty UTCSTAMP option (Section 3.4) in an EDNS OPT pseudo-RR in the query message.

The resolver MUST NOT include any UTCSTAMP payload data in the query message.

The resolver MUST NOT expect the server to respond with a valid UTCSTAMP response.

The resolver MUST be able to handle the situation that the request is ignored by the server.

The resolver SHOULD be able to handle unsolicited UTCSTAMP data sent by the server.

3.3. Name Server Behavior

A name server that understands the UTCSTAMP option, and chooses to honour a particular UTCSTAMP request, responds by including time information in a UTCSTAMP option (Section 3.4) in an EDNS OPT pseudo-RR in the response message.

The name server MUST ignore any UTCSTAMP payload data that might be present in the query message.

A name server MUST NOT send a UTCSTAMP option back to a resolver unless it was requested.

The UTCSTAMP option is not transitive. In particular, a recursive name server MUST consider UTCSTAMP incoming transactions its client as a separate from its outgoing transactions towards authoritative servers and/or forward resolvers. An incoming UTCSTAMP received from an authoritative (or forward) server MUST NOT be forwarded in an outgoing response to a client.

If a server doesn't have an understanding of the current time, it MUST either ignore the request, OR signal the fact that it cannot honour the request by responding with NOTIME data (see Section 3.4).

3.4. The UTCSTAMP Option

The OPTION-CODE for the UTCSTAMP option is 65394 (provisional and experimental, a permanent one to be assigned by the IANA, should this document be adopted as an RFC.).

OPTION-LEN MUST always be zero (0) in UTCSTAMP requests and MUST be at least eight (8) in responses. UTCSTAMP options with OPTION-LENs that deviate from these rules MUST be ignored.

The OPTION-DATA section for a UTCSTAMP request must be empty. For responses the UTCSTAMP OPTION-DATA is always an unsigned integer, in network byte order, at least 64-bits long, which represents the number of seconds since 1970-01-01 00:00:00 UTC, with the exception of the value with all bits set (i.e., FFFFFFFFFFFFFFFF (HEX) for a 64-bit number) which SHOULD be sent and received as a signal that the server giving out the information understands the request but is incapable of providing the information. The signal is referred to as "NOTIME". If the server understands the request and is able to provide the requested information, but is unwilling to do so (typically due to its configuration), the incoming UTCSTAMP request SHOULD be ignored, and no UTCSTAMP option response given at all.

3.5. Presentation Format

User interfaces SHOULD present the UTCSTAMP information using human readable ISO 8601 format and UTC time: "YYYY-MM-DD HH:MM:SS UTC" (e.g., "2018-11-15 15:52:23 UTC"). In addition to this, user interfaces MAY also offer to provide the compact ISO 8601 format: "YYYYMMDDTHHMMSSZ" (e.g., "20181115T155223Z"), and/or the pure data from the OPTION-DATA section of the message, in decimal notation (e.g., "1542383543").

The first format is preferred, as UTCSTAMP is intended as a debugging tool for humans, and that version is easier for humans to read. The latter may be useful in automated monitoring scenarios.

Software that present this information should examine data carefully, before processing it, and should not assume that the data is neither correct nor sensible.

The UTCSTAMP payload is binary data. Any automated comparison between UTCSTAMP payloads SHOULD be a comparison of the raw binary data or the raw data converted to the native byte order for the machine in question. Copy operations MUST NOT assume that the raw UTCSTAMP payload is null-terminated.

4. Discussion

This section discusses certain aspects of the protocol and explains considerations that led to the chosen design.

4.1. Data Payload

The choice of "epoch time" (number of seconds since 1970-01-01 00:00:00 UTC) was based on simplicity. Epoch time is a simple integer, that will fit in fixed 64-bit container for a long time to come. Epoch time needs no meta information, such as day of week or time zone, and there are no options or alternatives associated with epoch time. By eliminating options and alternatives, there is very little chance of misinterpretation on the recipient side, and since it is a simple number, comparison with other similar timestamps is very straightforward. In certain cases, which are arguably common, it also imposes very little need for computation at the server end.

There are of course arguments for making it more complex. An obvious one is that the server may not have, or be able to compute, the specified value. It may have no notion what so ever of a time zone. It may have an internal format that is based on a totally different format than epoch seconds, and may therefore have to perform some computations in order to produce the value.

For the former case, time comparison is of more limited value. If the server doesn't even know which time zone it sits in, it has no notion of universal time, and universal time is needed for timestamped signatures to work universally. There may be cases where a limited cluster of servers share a common understanding of time which is not based on time zones, but this protocol is intended to be generic and possible to use on the public global Internet.

For the case with a different internal time system, computation will indeed be necessary, but the authors argue that that case is less common, and that the computations are only moderately complex.

4.2. Presentation format

The presentation format was chosen based on readability and international standards. This protocol is primarily intended to be used by humans when debugging DNS systems. The international standard ISO 8601 specifies different ways to express time and date. The chosen model is a common version of ISO 8601 "extended version", where standardised punctuation is utilised to facilitate readability. It retains the important property that the order of significance is monotonously decreasing. The most significant value (the year) is to the far left, and the least significant value (seconds) is to the far right. This makes it easy for humans to compare two dates.

The alternative forms suggested are intended for programmatic use, and may be easier for computers to parse.

5. IANA Considerations

This memo includes a request to the IANA to assign an EDNS option code for UTCSTAMP.

6. Security Considerations

This protocol is intended as and aid in debugging scenarios. If this protocol signals that the server's notion of current time differs significantly from that of the client, that is an indicator of a possible problem. If the time given by the server matches that of the client, no knowledge has been gained.

This protocol is not intended to be a way to obtain trustworthy time information. There is no guarantee that the server responds with correct data, and the transport of the result is questionable. The transport of data can normally be secured by using TSIG, but as a logical somersault, the primary intent of the protocol is to aid in debugging scenarios where TSIG doesn't work. Therefore TSIG cannot be depended on to provide secure transport.

This protocol SHOULD NOT be used to try obtain correct time, as there is no way to ensure that the information is correct. Trusting time obtained with this protocol may lead to complex fault scenarios where incorrect time makes signature validation fail - a situation that can be very difficult to get out without manual intervention - or scenarios where replay attack may succeed.

This protocol is primarily intended to be used to compare time between two DNS parties.

7. Change History

draft-liman-dns-utcstamp-00

Initial version. Released 2018-11-20.

8. Document Timestamp

(This is support for the author. To be removed before publication.)

Document Time-stamp: "2018-11-20 16:10:42 liman"

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

9.2. Informative References

[RFC5001] Austein, R., "DNS Name Server Identifier (NSID) Option", RFC 5001, DOI 10.17487/RFC5001, August 2007, <https://www.rfc-editor.org/info/rfc5001>.

Authors' Addresses

Lars-Johan Liman
Netnod
Box 30194
SE-104 25 Stockholm
SE

Phone: +46 8 562 860 00
Email: liman@netnod.se
URI: https://www.netnod.se/

Ragnar Sundblad
Royal Institute of Technology
SE-100 44 Stockholm
SE

Phone: +46 8 790 60 00
Email: ragge@kth.se
URI: https://www.kth.se/