                Network Time Protocol I-Do Extension Field
                        draft-stenn-ntp-i-do-05

Abstract

   The first implementation of NTPv4 was released in 2003. NTPv4 is
   defined by RFC 5905 [RFC5905]. It contains a public-key security
   protocol, Autokey, which is defined by RFC 5906 [RFC5906]. Until
   very recently, Autokey has been the only defined "user" of NTP packet
   Extension Fields. New proposals for extension fields are being
   written and there is currently no convenient way to learn if a remote
   instance of NTP supports any extension fields or not. This proposal
   contains a method to tell a remote instance of NTP what we (are
   willing to admit we) support, and ask what they (are willing to admit
   they) support.

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 April 5, 2019.

Copyright Notice

carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must
include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction

The first implementation of NTPv4 was released in 2003. NTPv4 is
defined by RFC 5905 [RFC5905]. It contains a public-key security
protocol, Autokey, which is defined by RFC 5906 [RFC5906]. Until
very recently, Autokey has been the only defined "user" of NTP packet
Extension Fields. New proposals for extension fields are being
written and there is currently no convenient way to learn if a remote
instance of NTP supports any extension fields or not. This proposal
contains a method to tell a remote instance of NTP what we (are
willing to admit we) support, and ask what they (are willing to admit
they) support.

1.1. 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].

2. The I-Do Extension Field

The purpose of the I-DO EF is to provide information to the remote
side about our capabilities.

If an incoming packet contains an unrecognized extension field, one
of several things will happen. While that unrecognized extension
field SHOULD be ignored, an implementation MAY choose to drop the
entire packet. If any extension field is present there ordinarily
SHOULD be a MAC following the extension field, but an older
conforming NTP implementation would assume that any EF MUST be
followed by a MAC. Some extension fields are unable to be "signed"
by a MAC, regardless of whether or not that MAC is a traditional MAC

or an extension field MAC. In the final case, the receiving system
will interpret the unrecognized EF as a legacy MAC, and return a
crypto-NAK.

If the remote system replies with a crypto-NAK, that is a good
indication that it is running older software that does not recognize
EFs and thinks we have sent an invalid MAC. In this case, we should
behave accordingly with regard to the remote system.

If the remote system replies without including an I-DO-RESPONSE EF,
we at least know they can handle EFs, but they either don't
understand I-DO or are not willing to tell us anything.

If the remote system replies with a packet that includes an I-DO-
RESPONSE EF, then we SHOULD remember what they told us, and use that
information appropriately.

In client/server mode, it makes sense for the client to send an I-DO
to the server, and notice how the server responds. It likely does
not make sense for the server to send an I-DO EF in response to a
client request.

In symmetric mode, either side may initiate sending an I-DO EF, and
the receiving side SHOULD reply with an I-DO-RESPONSE EF.

In broadcast mode, the broadcast server MAY send broadcast packets
that include an I-DO EF, but note that if, counter to recommended
practice, these packets are unauthenticated they MAY cause client
machines to misinterpret the packet as having invalid authentication.
In this situation, the broadcast server SHOULD alternate sending
broadcast server packets with and without an I-DO EF, to insure that
all clients receive time packets they will accept. Note that if, as
recommended, broadcast packets are authenticated, a conforming client
SHOULD have no difficulty in receiving a broadcast (mode 5) packet
from a server that includes an I-DO EF.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-----------------------------+
|           Field Type          |           Field Length      |
+-----------------------------+-------------------------------+
|            I-Do 1           |              ...             |
+-----------------------------+-------------------------------+
|            I-Do N           |             Padding          |
+-------------------------------------------------------------+
```

NTP Extension Field: REFID Suggestion

Field Type: TBD (Recommendation for IANA: 0x0007 (I-Do), 0x8007 (I-Do
Response))

Field Length: as needed

Payload: An enumeration of the supported base Field Types, followed
by any padding, 0x0000, needed to fill the payload to the desired
32-bit boundary.

Example: A system that wants to advertise support for Autokey and
I-Do, sending to a system that responds with support for I-Do, NTS,
and MAC-As-Extension-Field

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-----------------------------+
|    Field Type (0x0007)        |    Field Length (0x0008)    |
+-------------------------------+-----------------------------+
|              0x0007           |              0x0002         |
+-------------------------------+-----------------------------+
```

                    NTP Extension Field: I-Do

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-----------------------------+
|    Field Type (0x8007)        |    Field Length (0x000a)    |
+-------------------------------+-----------------------------+
|              0x0003           |              0x0004         |
+-------------------------------+-----------------------------+
|              0x0007           |              0x0000         |
+-------------------------------+-----------------------------+
```

                  NTP Extension Field: I-Do Response

The sender of any I-Do extension field MUST send an extension field
with a Field Type of 0x0007 (I-Do) and SHOULD include a payload with
any 0x0000 padding values after enumerating the supported base
Extension Field Types. If the responding system recognizes the I-Do
extension field, its response MUST include an extension field with a
Field Type of 0x8007 (I-Do Response), and SHOULD include a payload
with any 0x0000 padding values after enumerating the supported base
Extension Field Types.

Any system that receives an I-Do extension field as either an "offer"
or a "response" SHOULD scan the entire payload looking for nonzero
values that specify the capabilities of the remote association.

Any system that receives an I-Do "offer", 0x0007, SHOULD reply with
an I-Do "response", 0x8007.

Any system that sends an I-Do "offer" or "response" may send as few
or as many of its supported Field Types as it chooses. At any
subsequent time, either side may re-negotiate the list of supported
field types it is prepared to accept from the other system by sending
a new I-Do extension field.

The most-recently received I-Do list replaces any previous I-Do list.

3. IANA Considerations

This memo requests IANA to allocate NTP Extension Field Types:

    0x0007 (I-DO)

    0x8007 (I-DO Response)

and I-DO types:

    0xFFFE (I-DO Leap Smear REFIDs)

    0xFFFF (I-DO IPv6 REFID hash)

for this proposal.

4. Security Considerations

Additional information TBD

5. 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>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch,
           "Network Time Protocol Version 4: Protocol and Algorithms
           Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010,
           <https://www.rfc-editor.org/info/rfc5905>.

[RFC5906] Haberman, B., Ed. and D. Mills, "Network Time Protocol
           Version 4: Autokey Specification", RFC 5906,
           DOI 10.17487/RFC5906, June 2010,
           <https://www.rfc-editor.org/info/rfc5906>.

Author's Address

    Harlan Stenn
    Network Time Foundation
    P.O. Box 918
    Talent, OR 97540
    US

    Email: stenn@nwtime.org