```
Internet Engineering Task Force                              H. Stenn
Internet-Draft                              Network Time Foundation
Intended status: Standards Track                         S. Goldberg
Expires: April 7, 2019                             Boston University
                                                    October 4, 2018
```

Network Time Protocol REFID Updates
draft-ietf-ntp-refid-updates-04

Abstract

   RFC 5905 [RFC5905], section 7.3, "Packet Header Variables", defines
   the value of the REFID, the system peer for the responding host. In
   the past, for IPv4 associations the IPv4 address is used, and for
   IPv6 associations the first four octets of the MD5 hash of the IPv6
   are used. There are two recognized shortcomings to this approach,
   and this proposal addresses them. One is that knowledge of the
   system peer is "abusable" information and should not be generally
   available. The second is that the four octet hash of the IPv6
   address looks very much like an IPv4 address, and this is confusing.

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 7, 2019.

Copyright Notice

Table of Contents

1. Introduction

1.1. The REFID

The interpretation of a REFID is based on the stratum, as documented
in RFC 5905 [RFC5905], section 7.3, "Packet Header Variables". The
core reason for the REFID in the NTP Protocol is to prevent a degree-
one timing loop, where server B decides to follow A as its time
source, and A then decides to follow B as its time source.

At Stratum 2+, which will be the case if two servers A and B are
exchanging timing information, then if server B follows A as its time
source, A's address will be B's REFID. When A uses IPv4, the default
REFID is A's IPv4 address. When A uses IPv6, the default REFID is a
four-octet digest of A's IPv6 address. Now, if A queries B for its
time, then A will learn that B is using A as its time source by
observing A's address in the REFID field of the response packet sent
by B. Thus, A will not select B as a potential time source, as this
would cause a timing loop.

1.2. NOT-YOU REFID

   The traditional REFID mechanism, however, also allows a third-party C
   to learn that A is the time source that is being used by B. When A
   is using IPv4, C can learn this by querying B for its time, and
   observing that the REFID in B's response is the IPv4 address of A.
   Meanwhile, when A is using IPv6, then C can again query B for its
   time, and then can use an offline dictionary attack to attempt to
   determine the IPv6 address that corresponds to the digest value in
   the response sent by B. C could construct the necessary dictionary
   by compiling a list of publicly accessible IPv6 servers. Remote
   attackers can use this technique to attempt to identify the time
   sources used by a target, and then send spoofed packets to the target
   or its time source in an attempt to disrupt time service, as was done
   e.g., in [NDSS16] or [CVE-2015-8138].

   The REFID thus unnecessarily leaks information about a target's time
   server to remote attackers. The best way to mitigate this
   vulnerability is to decouple the IP address of the time source from
   the REFID. To do this, a system can use an otherwise-impossible
   value for its REFID, called the "not-you" value, when it believes
   that a querying system is not its time source.

   The NOT-YOU REFID proposal is backwards-compatible. It can be
   implemented by one peer in an NTP association without any changes to
   the other peer.

   The NOT-YOU REFID proposal does have a small risk, in that a system
   that might return NOT-YOU does not have perfect information and it is
   possible that the remote system peer is contacting "us" via a
   different network interface. In this case, the remote system might
   choose us as their system peer, and a degree-one timing loop will
   occur. In this case, however, the two systems will spiral into
   worsening stratum positions with increasing root distances, and
   eventually the loop will break. If any other systems are available
   as time servers, one of them may become the new system peer.
   However, unless or until this happens the two spiraling systems will
   have degraded time quality.

1.3. IPv6 REFID

   In an environment where all time queries made to a server can be
   trusted, an operator might well choose to expose the real REFID. RFC
   5905 [RFC5905], section 7.3, "Packet Header Variables", explains how
   a remote system peer is converted to a REFID. It says:

If using the IPv4 address family, the identifier is the four-octet
IPv4 address. If using the IPv6 family, it is the first four
octets of the MD5 hash of the IPv6 address. ...

However, the MD5 hash of an IPv6 address often looks like a valid
IPv4 address. When this happens, an operator cannot tell if the
REFID refers to an IPv6 address or and IPv4. Specifically, the NTP
Project has received a report where the generated IPv6 hash decoded
to the IPv4 address of a different machine on the system peer's
network.

This proposal offers a way for a system to generate a REFID for a
IPv6 system peer that does not conflict with an IPv4-based REFID.

This proposal is not fully backwards-compatible. It SHOULD be
implemented by both peers in an NTP association. In the scenario
where A and B are peering using IPv6, where A is the system peer and
does not understand IPv6 REFID, and B is subordinate and is using
IPv6 REFID, A will not be able to determine that B is using A as its
system peer and a degree-one timing loop can form.

If both peers implement the IPv6 REFID this situation cannot happen.

[If at least one of the peers implements the proposed I-DO protocol
this situation cannot happen.]

1.4. 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 NOT-YOU REFID

2.1. Proposal

When enabled, this proposal allows the one-degree loop detection to
work and useful diagnostic information to be provided to trusted
partners while keeping potentially abusable information from being
disclosed to ostensibly uninterested parties. It does this by
returning the normal REFID to queries that come from trusted
addresses or from an address that the current system believes is its
time source (aka its "system peer"), and otherwise returning one of
two special IP addresses that is interpreted to mean "not you". The
"not you" IP addresses are 127.127.127.127 and 127.127.127.128. If
an IPv6 query is received from an address whose four-octet hash
equals one of these two addresses and we believe the querying host is

not our system peer, the other NOT-YOU address is returned as the
REFID.

This mechanism is correct and transparent when the system responding
with a NOT-YOU can accurately detect when it's getting a timing query
from its system peer. A querying system that uses IPv4 continues to
check that its IPv4 address does not appear in the REFID before
deciding whether to take time from the current system. A querying
system that uses IPv6 continues to check that the four-octet hash of
its IPv6 address does not appear in the REFID before deciding whether
to take time from the current system. However...

Use of the NOT-YOU REFID proposal will hide the current system's
system peer from querying systems that the current system believes
are not the current system's system peer. Should the current system
return the "not you" REFID to a query from its system peer, for
example in the case where the system peer sends its query from an
unexpected IP address, a one-degree timing loop can occur. Put
another way, the responding system has imperfect knowledge about
whether or not the sender is its system peer and there are cases
where it will offer a NOT-YOU response to its system peer, which can
then produce a degree-one timing loop.

Note that this mechanism fully supports degree-one loop detection in
the case where the responding NOT-YOU system can accurately detect
when it's getting a request from its system peer, and otherwise
provides the most basic diagnostic information to third parties.

3. Augmenting the IPv6 REFID Hash

3.1. Background

In a trusted network, the S2+ REFID is generated based on the network
system peer. RFC 5905 [RFC5905] says:

   If using the IPv4 address family, the identifier is the four-octet
   IPv4 address. If using the IPv6 family, it is the first four
   octets of the MD5 hash of the IPv6 address. ...

This means that the IPv4 representation of the IPv6 hash would be:
b1.b2.b3.b4 . This proposal is that the system MAY also use
255.b2.b3.b4 as its REFID. This reduces the risk of ambiguity, since
addresses beginning with 255 are "reserved", and thus will not
collide with valid IPv4 on the network.

When using the REFID to check for a timing loop for an IPv6
association, if the code that checks the first four-octets of the

hash fails to match then the code must check again, using 0xFF as the first octet of the hash.

3.2. Potential Problems

There is a 1 in 16,777,216 chance that the REFID hashes of two IPv6 addresses will be identical, producing a false-positive loop detection. With a sufficient number of servers, the risk of this problem becomes a non-issue. [The use of the NOT-YOU REFID and/or the proposed "REFID Suggestion" or "I-DO" extension fields are ways to mitigate this potential situation.]

Unrealistically, if only two instances of NTP are communicating via IPv6 and system A implements this new IPv6 REFID hash and system B does not, system B will not be able to detect this loop condition. In this case, the two machines will slowly increase their Stratum until they reach S16 and become unsynchronized. This situation is considered to be unrealistic because, for this to happen, each system would have to have only the other system available as a time source, for example, in a misconfigured "orphan mode" setup. There is no risk of this happening in an NTP network with 3 or more time sources, or in a properly-configured "time island" setup.

3.3. Questions

Should we reference the REFID Suggestion and I-DO proposals here?

4. Acknowledgements

For the "not-you" REFID, we acknowledge useful discussions with Aanchal Malhotra and Matthew Van Gundy.

For the IPv6 REFID, we acknowledge Dan Mahoney (and perhaps others) for suggesting the idea of using an "impossible" first-octet value to indicate an IPv6 refid hash.

5. IANA Considerations

This memo requests IANA to allocate a pseudo Extension Field Type of 0xFFFF so the proposed "I-Do" exchange can report whether or not the "IPv6 REFID Hash" is supported.

6. Security Considerations

Many systems running NTP are configured to return responses to timing queries by default. These responses contain a REFID field, which generally reveals the address of the system's time source if that source is an IPv4 address. This behavior can be exploited by remote

attackers who wish to first learn the address of a target's time
source, and then attack the target and/or its time source. As such,
the "not-you" REFID proposal is designed to harden NTP against these
attacks by limiting the amount of information leaked in the REFID
field.

Systems running NTP should reveal the identity of their system in
peer in their REFID only when they are on a trusted network. The
IPv6 REFID proposal provides one way to do this, when the system peer
uses addresses in the IPv6 family.

7. References

7.1. 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>.

[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>.

7.2. Informative References

[CVE-2015-8138]
            Van Gundy, M. and J. Gardner, "Network Time Protocol
            Origin Timestamp Check Impersonation Vulnerability (CVE-
            2015-8138)", in TALOS VULNERABILITY REPORT (TALOS-
            2016-0077), 2016.

[NDSS16]   Malhotra, A., Cohen, I., Brakke, E., and S. Goldberg,
            "Attacking the Network Time Protocol", in ISOC Network and
            Distributed System Security Symposium 2016 (NDSS'16),
            2016.

Authors' Addresses

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

    Email: stenn@nwtime.org

Sharon Goldberg
Boston University
111 Cummington St
Boston, MA 02215
US

Email: goldbe@cs.bu.edu