Internet Engineering Task Force H. Stenn
Internet-Draft Network Time Foundation
Intended status: Standards Track S. Goldberg
Expires: December 8, 2018 Boston University
June 6, 2018

Network Time Protocol REFID Updates
draft-ietf-ntp-refid-updates-03

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 at least three shortcomings to this approach, and this proposal will address the three so noted. 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. The third is that a growing number of low-stratum servers want to offer leap-smeared time to their clients, and there is no obvious way to know if a server is offering accurate time or leap-smeared time.

Status of This Memo

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, since this would cause a timing loop.

1.2. NOT-YOU REFID

This 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 REFID value, when it believes that a querying system is not its time source.

The NOT-YOU REFID proposal is backwards-compatible and provides the most basic diagnostic information to third parties. It can be implemented by one peer in an NTP association without any changes to the other peer. This holds as long as responding NOT-YOU system can accurately detect when it's getting a request from its system peer.

The NOT-YOU REFID proposal does have a small risk. Consider system A that returns the NOT-YOU REFID and system B that has two network interfaces B1 and B2. Suppose that system A is using system B as his time source, via network interface B1. Now suppose that system B queries system A for time via network interface B2. In this case, system A returns the NOT-YOU REFID value to system B, since system A does not realize that network interface B1 and B2 belong to the same system. In this case, system B might choose system A as its time source, and a degree-one timing loop will occur. In this case, however, the two systems will spiral into worse stratum positions with increasing root distances, and eventually the loop will break.

If any other systems are available as time servers, one of them will become the new system peer. However, 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 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. Leap-Smear REFID

RFC 5905 [RFC5905] and earlier versions of NTP are the overwhelming method of distributing time on networks. Leap Seconds will continue to exist for a good number of years' time, and since the timescale mandated by POSIX effectively ignores any instances where there are not 86,400 seconds' time in a day something must be done to reliably synchronize clocks during the application of leap second corrections. One way to deal with the insertion of a leap second is to apply the leap second using a "smear", where the time reported by leap-second

aware servers is gradually adjusted so there is no major disruption to time synchronization when processing a leap second.

While the proper handling of leap seconds can be expected from up-to-date software and time servers, there are large numbers of out-of-date software installations and systems that are just not able to properly handle a leap second correction.

This proposal offers a way for a system to generate a REFID that indicates that the time being supplied in the NTP packet already contains an amount of leap smear correction, and what that amount is.

This proposal is backwards-compatible in all but poorly-designed NTP networks. The entire point of providing NTP servers that offer leap-smeared time in response to CLIENT requests is to provide smooth time to clients that are unable to properly handle leap seconds. If an operator is skilled enough to provide leap-smeared time to a subset of clients that cannot properly handle leap seconds, they can be expected to know enough to avoid using leap-smeared time between time servers that are expected to be able to properly handle leap seconds. Leap smears are expected to be implemented on a limited number of time servers where there is a base of client systems that cannot handle a leap second correction. Furthermore, even in a poorly-designed NTP network the "window of risk" lasts only as long as it takes for the leap second to be smeared.

1.5. 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 a special IP address that is interpreted to mean "not you". The "not you" IP address is 127.127.127.127 when the query is made from an IPv4 address, or when the query is made from an IPv6 address whose four-octet hash does not equal 127.127.127.127. The "not you" IP

address is 127.127.127.128 when the query is made from an address whose four-octet hash equals 127.127.127.127.

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.

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 . The 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 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?

Should we ask IANA to allocate a pseudo Extension Field Type of 0xFFFF (for example) so the proposed "I-Do" exchange can report whether or not the "IPv6 REFID Hash" is supported?

4. The REFID sent to clients during a Leap-Smear

4.1. Background

This proposal offers a way for a system to generate a REFID that indicates that the time being supplied in the NTP packet already contains an amount of leap smear correction, and what that amount is.

4.2. Leap Smear REFID

RFC 5905 [RFC5905] defines the data type of NTP time values in Section 6, "Data Types":

> All NTP time values are represented in twos-complement format, with bits numbered in big-endian (as described in Appendix A of [RFC0791]) fashion from zero starting at the left, or high-order, position. ...

The 32 bit signed integer seconds portion and the 32 bit unsigned fractional seconds portion, or 32:32 format is:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Seconds                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Fraction                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

NTP Timestamp Format (32:32)

This format provides coverage for 136 years' time to a precision of 232 picoseconds. If a leap-second addition is being completely smeared just before before the stroke of the next POSIX second then

the smear correction will be (0,1). [That's mathematical domain range notation - how to cite it?] If this was the only way to apply a leap smear correction then we could simply use an unsigned value to represent the correction. But while the first popular leap smear implementation applied the correction over an appropriate number of hours' time before the actual leap second, so the system time was again correct at the stroke of 00:00, that meant that the difference between system time and UTC spent half of the duration of the smear application at [.5,1) "off" of correct time. The second popular implementation of the leap smear applied the first half-second correction before the stroke of 00:00 for a correction range of (0,.5] and the last half-second correction starting at the stroke of 00:00 for a [-.5,0) correction range. This also means we need a signed value to represent the amount of correction.

The REFID of a system that is supplying smeared time to client requests while leap-smear correction is active would be 254.b1.b2.b3, where the three octets (b1, b2, and b3) are a 2:22 formatted value, yielding 2 signed bits of integer time and 22 bits of unsigned fractional subseconds, with a precision to 238 nanoseconds, or about a quarter of a microsecond. Signed time is needed to implement the mathematical range described in the previous paragraph.

[How should we cite the 2:22 notation? This is the same general format that we use for NTP timestamps.]

[Sharon says: I suggest adding a concrete example of the scheme, so that the above paragraph is easier to understand.]

The client is not expected to do anything with this information. Indeed, the whole point of offering smeared time is that there is reason to believe the clients are unable to properly handle a leap second correction. In this case, clients cannot be expected to do anything with data embedded in the REFID, either. However, monitoring systems that use tools that show a host's system peer, like the 'ntpq' and 'sntp' programs in the reference implementation, [HMS: how to cite this?] can use this information to make sure that clients are following a leap-smearing server and can see fairly accurately what the smear is for each client.

Note that if an NTP server decides to offer smeared time corrections to clients, it SHOULD only offer this time in response to CLIENT time requests. An NTP server that is offering smeared time SHOULD NOT send smeared time in any peer exchanges. Also, system that sync their time via CLIENT requests SHOULD NOT be distributing time (smeared or otherwise) to other systems.

[Sharon asks: Consider a client that doesn't know he is getting smeared time (b\c he is outdated etc). How is a this client supposed to know that he should not be distributing smeared time? Note that its perfectly normal for a stratum 2 server that gets his time via CLIENT requests from a stratum 1 server to then offer time to stratum 3 systems.]

We also note that during the application of a leap smear, the REFID from a system offering smeared time cannot provide detection of a timing loop. This is not expected to be a problem because time server systems are not expected to make CLIENT connections with each other, so they should not be receiving smeared time. [Sharon asks: I don't understand this point, see my question above.] Moreso, if a time server is configured to make CLIENT connections to a server that offers smeared time, with the mechanism described here it can detect when it is getting smeared time, and either ignore time from that source, or "undo" the leap smear correction and use the corrected time for that sample.

This proposal is not an attempt to justify servers offering leap smeared time. It is only an attempt to make it easy and visible to identify when a server is offering or client is receiving smeared time, and provide the client a means to know the amount of smear correction as of the latest successful poll.

4.3. Questions

Should we ask IANA to allocate a pseudo Extension Field Type of 0xFFFE (for example) so the proposed "I-Do" exchange can report whether or not this server will offer leap smeared time in response to CLIENT time requests, identifying the amount of correction using the above REFID?

5. 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.

For the Leap Smear REFID, we acknowledge useful discussions with Juergen Perlinger.

6. IANA Considerations

This memo makes no requests of IANA.

7. 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.

8. References

8.1. 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>.

8.2. 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