Network Time Protocol (ntp) Working Group
Internet-Draft
Updates: 5905 (if approved)
Intended status: Standards Track
Expires: December 12, 2021

F. Gont
G. Gont
SI6 Networks
M. Lichvar
Red Hat

June 10, 2021

Port Randomization in the Network Time Protocol Version 4
draft-ietf-ntp-port-randomization-08

Abstract

The Network Time Protocol can operate in several modes. Some of these modes are based on the receipt of unsolicited packets, and therefore require the use of a well-known port as the local port number. However, in the case of NTP modes where the use of a well-known port is not required, employing such well-known port unnecessarily facilitates the ability of attackers to perform blind/off-path attacks. This document formally updates RFC5905, recommending the use of transport-protocol ephemeral port randomization for those modes where use of the NTP well-known port is not required.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Network Time Protocol (NTP) is one of the oldest Internet protocols, and currently specified in [RFC5905]. Since its original implementation, standardization, and deployment, a number of vulnerabilities have been found both in the NTP specification and in some of its implementations [NTP-VULN]. Some of these vulnerabilities allow for off-path/blind attacks, where an attacker can send forged packets to one or both NTP peers for achieving Denial of Service (DoS), time-shifts, or other undesirable outcomes. Many of these attacks require the attacker to guess or know at least a target NTP association, typically identified by the tuple {srcaddr, srcport, dstaddr, dstport, keyid} (see section 9.1 of [RFC5905]). Some of these parameters may be easily known or guessed.

NTP can operate in several modes. Some of these modes rely on the ability of nodes to receive unsolicited packets, and therefore require the use of the NTP well-known port (123). However, for modes where the use of a well-known port is not required, employing the NTP well-known port unnecessarily facilitates the ability of an attacker to perform blind/off-path attacks (since knowledge of the port

numbers is typically required for such attacks). A recent study [NIST-NTP] that analyzes the port numbers employed by NTP clients suggests that a considerable number of NTP clients employ the NTP well-known port as their local port, or select predictable ephemeral port numbers, thus unnecessarily facilitating the ability of attackers to perform blind/off-path attacks against NTP.

BCP 156 [RFC6056] already recommends the randomization of transport-protocol ephemeral ports. This document aligns NTP with the recommendation in BCP 156 [RFC6056], by formally updating [RFC5905] such that port randomization is employed for those NTP modes for which the use of the NTP well-known port is not needed.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Considerations About Port Randomization in NTP

The following subsections analyze a number of considerations about transport-protocol ephemeral port randomization when applied to NTP.

3.1. Mitigation Against Off-path Attacks

There has been a fair share of work in the area of off-path/blind attacks against transport protocols and upper-layer protocols, such as [RFC5927] and [RFC4953]. Whether the target of the attack is a transport protocol instance (e.g., TCP connection) or an upper-layer protocol instance (e.g., an application protocol instance), the attacker is required to know or guess the five-tuple {Protocol, IP Source Address, IP Destination Address, Source Port, Destination Port} that identifies the target transport protocol instance or the transport protocol instance employed by the target upper-layer protocol instance. Therefore, increasing the difficulty of guessing this five-tuple helps mitigate blind/off-path attacks.

As a result of these considerations, transport-protocol ephemeral port randomization is a best current practice (BCP 156) that helps mitigate off-path attacks at the transport-layer. This document aligns the NTP specification [RFC5905] with the existing best current practice on ephemeral port selection, irrespective of other techniques that may (and should) be implemented for mitigating off-path attacks.

We note that transport-protocol ephemeral port randomization is a transport-layer mitigation against off-path/blind attacks, and does not preclude (nor is it precluded by) other possible mitigations for off-path attacks that might be implemented at other layers (e.g. [I-D.ietf-ntp-data-minimization]). For instance, some of the aforementioned mitigations may be ineffective against some off-path attacks [NTP-FRAG] or may benefit from the additional entropy provided by port randomization [NTP-security].

3.2. Effects on Path Selection

Intermediate systems implementing the Equal-Cost Multi-Path (ECMP) algorithm may select the outgoing link by computing a hash over a number of values, that include the transport-protocol source port. Thus, as discussed in [NTP-CHLNG], the selected client port may have an influence on the measured offset and delay.

If the source port is changed with each request, packets in different exchanges will be more likely to take different paths, which could cause the measurements to be less stable and have a negative impact on the stability of the clock.

Network paths to/from a given server are less likely to change between requests if port randomization is applied on a per-association basis. This approach minimizes the impact on the stability of NTP measurements, but may cause different clients in the same network synchronized to the same NTP server to have a significant stable offset between their clocks due to their NTP exchanges consistently taking different paths with different asymmetry in the network delay.

Section 4 recommends NTP implementations to randomize the ephemeral port number of client/server associations. The choice of whether to randomize the port number on a per-association or a per-request basis is left to the implementation.

3.3. Filtering of NTP traffic

In a number of scenarios (such as when mitigating DDoS attacks), a network operator may want to differentiate between NTP requests sent by clients, and NTP responses sent by NTP servers. If an implementation employs the NTP well-known port for the client port number, requests/responses cannot be readily differentiated by inspecting the source and destination port numbers. Implementation of port randomization for non-symmetrical modes allows for simple differentiation of NTP requests and responses, and for the enforcement of security policies that may be valuable for the

mitigation of DDoS attacks, when all NTP clients in a given network employ port randomization.

3.4. Effect on NAPT devices

Some NAPT devices will reportedly not translate the source port of a packet when a system port number (i.e., a port number in the range 0-1023) [RFC6335] is employed. In networks where such NAPT devices are employed, use of the NTP well-known port for the client port may limit the number of hosts that may successfully employ NTP client implementations at any given time.

NOTES:

NAPT devices are defined in Section 4.1.2 of [RFC2663].

The reported behavior is similar to the special treatment of UDP port 500 that has been documented in Section 2.3 of [RFC3715].

In the case of NAPT devices that will translate the source port even when a system port is employed, packets reaching the external realm of the NAPT will not employ the NTP well-known port as the source port, as a result of the port translation function performed by the

NAPT device.

4. Update to RFC5905

The following text from Section 9.1 ("Peer Process Variables") of [RFC5905]:

> dstport: UDP port number of the client, ordinarily the NTP port number PORT (123) assigned by the IANA. This becomes the source port number in packets sent from this association.

is replaced with:

> dstport: UDP port number of the client. In the case of broadcast server mode (5) and symmetric modes (1 and 2), it SHOULD contain the NTP port number PORT (123) assigned by the IANA. In the client mode (3), it SHOULD contain a randomized port number, as specified in [RFC6056]. The value in this variable becomes the source port number of packets sent from this association. The randomized port number SHOULD NOT be shared with other associations, to avoid revealing the randomized port to other associations.
>
> If a client implementation performs ephemeral port randomization on a per-request basis, it SHOULD close the corresponding socket/port after each request/response exchange. In order to prevent

> duplicate or delayed server packets from eliciting ICMP port unreachable error messages at the client, the client MAY wait for more responses from the server for a specific period of time (e.g. 3 seconds) before closing the UDP socket/port.
>
> NOTES:
>
> Randomizing the ephemeral port number on a per-request basis will better mitigate off-path/blind attacks, particularly if the socket/port is closed after each request/response exchange, as recommended above. The choice of whether to randomize the ephemeral port number on a per-request or a per-association basis is left to the implementation, and should consider the possible effects on path selection along with its possible

impact on time measurement.

On most current operating systems, which implement ephemeral port randomization [RFC6056], an NTP client may normally rely on the operating system to perform ephemeral port randomization. For example, NTP implementations using POSIX sockets may achieve ephemeral port randomization by *not* binding the socket with the bind() function, or binding it to port 0, which has a special meaning of "any port". connect()ing the socket will make the port inaccessible by other systems (that is, only packets from the specified remote socket will be received by the application).

5. Implementation Status

[RFC Editor: Please remove this section before publication of this document as an RFC.]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

OpenNTPD:

[OpenNTPD] has never explicitly set the local port of NTP clients, and thus employs the ephemeral port selection algorithm implemented by the operating system. Thus, on all operating systems that implement port randomization (such as current versions of OpenBSD, Linux, and FreeBSD), OpenNTPD will employ port randomization for client ports.

chrony:

[chrony] by default does not set the local client port, and thus employs the ephemeral port selection algorithm implemented by the operating system. Thus, on all operating systems that implement port randomization (such as current versions of OpenBSD, Linux, and FreeBSD), chrony will employ port randomization for client ports.

nwtime.org's sntp client:
sntp does not explicitly set the local port, and thus employs the ephemeral port selection algorithm implemented by the operating system. Thus, on all operating systems that implement port randomization (such as current versions of OpenBSD, Linux, and FreeBSD), it will employ port randomization for client ports.

6. IANA Considerations

There are no IANA registries within this document. The RFC-Editor can remove this section before publication of this document as an RFC.

7. Security Considerations

The security implications of predictable numeric identifiers [I-D.irtf-pearg-numeric-ids-generation] (and of predictable transport-protocol port numbers [RFC6056] in particular) have been known for a long time now. However, the NTP specification has traditionally followed a pattern of employing common settings even when not strictly necessary, which at times has resulted in negative security and privacy implications (see e.g. [I-D.ietf-ntp-data-minimization]). The use of the NTP well-known port (123) for the srcport and dstport variables is not required for all operating modes. Such unnecessary usage comes at the expense of reducing the amount of work required for an attacker to successfully perform off-path/blind attacks against NTP. Therefore, this document formally updates [RFC5905], recommending the use of transport-protocol port randomization when use of the NTP well-known port is not required.

This issue has been assigned CVE-2019-11331 [VULN-REPORT] in the U.S. National Vulnerability Database (NVD).

8. Acknowledgments

The authors would like to thank (in alphabetical order) Ivan Arce, Roman Danyliw, Dhruv Dhody, Lars Eggert, Todd Glassey, Blake Hudson, Benjamin Kaduk, Erik Kline, Watson Ladd, Aanchal Malhotra, Danny Mayer, Gary E. Miller, Bjorn Mork, Hal Murray, Francesca Palombini, Tomoyuki Sahara, Zaheduzzaman Sarker, Dieter Sibold, Steven Sommars, Jean St-Laurent, Kristof Teichel, Brian Trammell, Eric Vyncke, Ulrich Windl, and Dan Wing, for providing valuable comments on earlier versions of this document.

Watson Ladd raised the problem of DDoS mitigation when the NTP well-known port is employed as the client port (discussed in Section 3.3 of this document).

The authors would like to thank Harlan Stenn for answering questions about nwtime.org's NTP implementation.

Fernando would like to thank Nelida Garcia and Jorge Oscar Gont, for their love and support.

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>.

[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>.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, DOI 10.17487/RFC6056, January 2011, <https://www.rfc-editor.org/info/rfc6056>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2. Informative References

[chrony] "chrony", <https://chrony.tuxfamily.org/>.

[I-D.ietf-ntp-data-minimization]
Franke, D. F. and A. Malhotra, "NTP Client Data Minimization", draft-ietf-ntp-data-minimization-04 (work in progress), March 2019.

[I-D.irtf-pearg-numeric-ids-generation]
Gont, F. and I. Arce, "On the Generation of Transient Numeric Identifiers", draft-irtf-pearg-numeric-ids-generation-07 (work in progress), February 2021.

[NIST-NTP]
Sherman, J. and J. Levine, "Usage Analysis of the NIST Internet Time Service", Journal of Research of the National Institute of Standards and Technology Volume 121, March 2016, <https://tf.nist.gov/general/pdf/2818.pdf>.

[NTP-CHLNG]
Sommars, S., "Challenges in Time Transfer Using the Network Time Protocol (NTP)", Proceedings of the 48th Annual Precise Time and Time Interval Systems and Applications Meeting, Monterey, California pp. 271-290, January 2017, <http://leapsecond.com/ntp/NTP_Paper_Sommars_PTTI2017.pdf>.

[NTP-FRAG]
Malhotra, A., Cohen, I., Brakke, E., and S. Goldberg, "Attacking the Network Time Protocol", NDSS'17, San Diego, CA. Feb 2017, 2017, <https://www.cs.bu.edu/~goldbe/papers/NTPattack.pdf>.

[NTP-security]
Malhotra, A., Van Gundy, M., Varia, V., Kennedy, H., Gardner, J., and S. Goldberg, "The Security of NTP's Datagram Protocol", Cryptology ePrint Archive Report 2016/1006, 2016, <https://eprint.iacr.org/2016/1006>.

[NTP-VULN]
Network Time Foundation, "Security Notice", Network Time Foundation's NTP Support Wiki ,
<https://support.ntp.org/bin/view/Main/SecurityNotice>.

[OpenNTPD]

"OpenNTPD Project", <https://www.openntpd.org>.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, DOI 10.17487/RFC2663, August 1999, <https://www.rfc-editor.org/info/rfc2663>.

[RFC3715] Aboba, B. and W. Dixon, "IPsec-Network Address Translation (NAT) Compatibility Requirements", RFC 3715, DOI 10.17487/RFC3715, March 2004, <https://www.rfc-editor.org/info/rfc3715>.

[RFC4953] Touch, J., "Defending TCP Against Spoofing Attacks", RFC 4953, DOI 10.17487/RFC4953, July 2007, <https://www.rfc-editor.org/info/rfc4953>.

[RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, DOI 10.17487/RFC5927, July 2010, <https://www.rfc-editor.org/info/rfc5927>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <https://www.rfc-editor.org/info/rfc7942>.

[VULN-REPORT] The MITRE Corporation, "CVE-2019-11331", April 2019, <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-11331>.

Authors' Addresses

Fernando Gont
SI6 Networks

Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: https://www.si6networks.com

Guillermo Gont
SI6 Networks
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: ggont@si6networks.com
URI: https://www.si6networks.com

Miroslav Lichvar
Red Hat
Purkynova 115
Brno 612 00
Czech Republic

Email: mlichvar@redhat.com