                 Network Time Protocol Best Current Practices

Abstract

   The Network Time Protocol (NTP) is one of the oldest protocols on the
   Internet and has been widely used since its initial publication.
   This document is a collection of best practices for the general
   operation of NTP servers and clients on the Internet. It includes
   recommendations for the stable, accurate, and secure operation of NTP
   infrastructure. This document is targeted at NTP version 4 as
   described in RFC 5905.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   NTP version 4 (NTPv4) has been widely used since its publication as
   [RFC5905]. This document is a collection of best practices for the
   operation of NTP clients and servers.

   The recommendations in this document are intended to help operators
   distribute time on their networks more accurately and securely. They
   are intended to apply generally to a broad range of networks. Some
   specific networks may have higher accuracy requirements that call for
   additional techniques beyond what is documented here.

   Among the best practices covered are recommendations for general
   network security, time protocol-specific security, and NTP server and
   client configuration. NTP operation in embedded devices is also
   covered.

   This document also contains information for protocol implementors who
   want to develop their own implementations compliant with RFC 5905.

1.1. Requirements Language

   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.

2. General Network Security Best Practices

2.1. BCP 38

   Many network attacks rely on modifying the IP source address of a
   packet to point to a different IP address than the computer from
   which it originated. UDP-based protocols, such as NTP, are generally
   more susceptible to spoofing attacks than connection-oriented
   protocols. NTP control messages can generate a lot of data in
   response to a small query, which makes it attractive as a vector for
   distributed denial-of-service attacks (NTP Control messages are
   discussed further in Section 3.4). One documented instance of such
   an attack can be found in [DDOS], with further discussion in [IMC14]
   and [NDSS14].

   BCP 38 [RFC2827] was published in 2000 to provide some level of
   remediation against address-spoofing attacks. BCP 38 calls for
   filtering outgoing and incoming traffic to make sure that the source
   and destination IP addresses are consistent with the expected flow of
   traffic on each network interface. It is RECOMMENDED that ISPs and

large corporate networks implement ingress and egress filtering.
More information is available at [BCP38WIKI].

3. NTP Configuration Best Practices

This section provides best practices for NTP configuration and
operation. Application of these best practices that are specific to
the Network Time Foundation implementation, including example
configuration directives valid at the time of this writing, are
compiled in Appendix A.

3.1. Keeping NTP Up to Date

There are multiple versions and implementations of the NTP protocol
in use on many different platforms. The practices in this document
are meant to apply generally to any implementation of [RFC5905]. NTP
users should select an implementation that is actively maintained.
Users should keep up to date on any known attacks on their selected
implementation and deploy updates containing security fixes as soon
as it is practical.

3.2. Using Enough Time Sources

An NTP implementation that is compliant with [RFC5905] takes the
available sources of time and submits this timing data to
sophisticated intersection, clustering, and combining algorithms to
get the best estimate of the correct time. The description of these
algorithms is beyond the scope of this document. Interested readers
should read [RFC5905] or the detailed description of NTP in
[MILLS2006].

   o If there is only one source of time, the answer is obvious. It
      may not be a good source of time, but it's the only source that
      can be considered. Any issue with the time at the source will be
      passed on to the client.

   o If there are two sources of time and they align well enough, then
      the best time can be calculated easily. But if one source fails,
      then the solution degrades to the single-source solution outlined
      above. And if the two sources don't agree, it will be difficult
      to know which one is correct without making use of information
      from outside of the protocol.

   o If there are three sources of time, there is more data available
      to converge on the best calculated time, and this time is more
      likely to be accurate. And the loss of one of the sources (by
      becoming unreachable or unusable) can be tolerated. But at that
      point, the solution degrades to the two-source solution.

   o Having four or more sources of time is better as long as the
      sources are diverse (Section 3.3). If one of these sources
      develops a problem, there are still at least three other time
      sources.

   This analysis assumes that a majority of the servers used in the
   solution are honest, even if some may be inaccurate. Operators
   should be aware of the possibility that if an attacker is in control
   of the network, the time coming from all servers could be
   compromised.

   Operators who are concerned with maintaining accurate time SHOULD use
   at least four independent, diverse sources of time. Four sources
   will provide sufficient backup in case one source goes down. If four
   sources are not available, operators MAY use fewer sources, which is
   subject to the risks outlined above.

   But even with four or more sources of time, systemic problems can
   happen. One example involves the leap-smearing concept detailed in
   Section 3.7.1. For several hours before and after the June 2015 leap
   second, several operators configured their NTP servers with leap
   smearing while others did not. Many NTP end nodes could not
   determine an accurate time source because two of their four sources
   of time gave them consistent UTC/POSIX time, while the other two gave
   them consistent leap-smeared time. This is just one of many
   potential causes of disagreement among time sources.

   Operators are advised to monitor all time sources that are in use.
   If time sources do not generally align, operators are encouraged to
   investigate the cause and either correct the problems or stop using
   defective servers. See Section 3.5 for more information.

3.3. Using a Diversity of Reference Clocks

   When using servers with attached hardware reference clocks, it is
   suggested that different types of reference clocks be used. Having a
   diversity of sources with independent implementations means that any
   one issue is less likely to cause a service interruption.

   Are all clocks on a network from the same vendor? They may have the
   same bugs. Even devices from different vendors may not be truly
   independent if they share common elements. Are they using the same
   base chipset? Are they all running the same version of firmware?
   Chipset and firmware bugs can happen, but they can be more difficult
   to diagnose than application software bugs. When having the correct
   time is of critical importance, it's ultimately up to operators to
   ensure that their sources are sufficiently independent, even if they
   are not under the operator's control.

A systemic problem with time from any satellite navigation service is
possible and has happened. Sunspot activity can render a satellite
or radio-based time source unusable. Depending on the application
requirements, operators may need to consider backup scenarios in the
rare circumstance when the satellite system is faulty or unavailable.

3.4. Control Messages

Some implementations of NTPv4 provide the NTP control messages (also
known as Mode 6 messages). These messages were originally specified
in Appendix B of [RFC1305], which defined NTPv3. These messages do
not appear in the NTPv4 specification [RFC5905], which obsoletes the
NTPv3 specification [RFC1305], but they are still used. At the time
of this writing, work is being done to formally document the
structure of these control messages for use with NTPv4 in [CTRLMSG].

NTP control messages are designed to permit monitoring and optionally
authenticated control of NTP and its configuration. Used properly,
these facilities provide vital debugging and performance information
and control. But these facilities can be a vector for amplification
attacks when abused. For this reason, it is RECOMMENDED that public-
facing NTP servers block NTP control message queries from outside
their organization.

The ability to use NTP control messages beyond their basic monitoring
capabilities SHOULD be limited to authenticated sessions that provide
a 'controlkey'. It can also be limited through mechanisms outside of
the NTP specification, such as Access Control Lists, that only allow
access from approved IP addresses.

The NTP control message responses are much larger than the
corresponding queries. Thus, they can be abused in high-bandwidth
DDoS attacks. Section 2.1 gives more information on how to provide
protection for this abuse by implementing BCP 38.

3.5. Monitoring

Operators SHOULD use their NTP implementation's remote monitoring
capabilities to quickly identify servers that are out of sync and
ensure correct functioning of the service. Operators SHOULD also
monitor system logs for messages so that problems and abuse attempts
can be quickly identified.

If a system starts to receive NTP Reply packets from a remote time
server that do not correspond to any requests sent by the system,
that can be an indication that an attacker is forging that system's
IP address in requests to the remote time server. The goal of this
attack is to adversely impact the availability of time to the

targeted system whose address is being forged. Based on these forged
packets, the remote time server might decide to throttle or rate-
limit packets or even stop sending packets to the targeted system.

If a system is a broadcast client and its system log shows that it is
receiving early time messages from its server, that is an indication
that somebody may be forging packets from a broadcast server
(broadcast client and server modes are defined in Section 3 of
[RFC5905]).

If a server's system log shows messages that indicate it is receiving
NTP timestamps that are much earlier than the current system time,
then either the system clock is unusually fast or somebody is trying
to launch a replay attack against that server.

3.6. Using Pool Servers

It only takes a small amount of bandwidth and system resources to
synchronize one NTP client, but NTP servers that can service tens of
thousands of clients take more resources to run. Network operators
and advanced users who want to synchronize their computers MUST only
synchronize to servers that they have permission to use.

The NTP Pool Project is a group of volunteers who have donated their
computing and bandwidth resources to freely distribute time from
primary time sources to others on the Internet. The time is
generally of good quality but comes with no guarantee whatsoever. If
you are interested in using this pool, please review their
instructions at [POOLUSE].

Vendors can obtain their own subdomain that is part of the NTP Pool
Project. This offers vendors the ability to safely make use of the
time distributed by the pool for their devices. Details are
available at [POOLVENDOR].

If there is a need to synchronize many computers, an operator may
want to run local NTP servers that are synchronized to the NTP Pool
Project. NTP users on that operator's networks can then be
synchronized to local NTP servers.

3.7. Leap-Second Handling

UTC is kept in agreement with the Universal Time UT1 [SOLAR] to
within +/- 0.9 seconds by the insertion (or possibly deletion) of a
leap second. UTC is an atomic time scale, whereas UT1 is based on
the rotational rate of the earth. Leap seconds are not introduced at

a fixed rate. They are announced by the International Earth Rotation
and Reference Systems Service (IERS) in its Bulletin C [IERS] when
necessary to keep UTC and UT1 aligned.

NTP time is based on the UTC timescale, and the protocol has the
capability to broadcast leap-second information. Some global
navigation satellite systems (like GPS) or radio transmitters (like
DCF77) broadcast leap-second information. If an NTP client is synced
to an NTP server that provides leap-second notification, the client
will get advance notification of impending leap seconds
automatically.

Since the length of the UT1 day generally slowly increases [SOLAR],
all leap seconds that have been introduced since the practice started
in 1972 have been positive leap seconds, where a second is added to
UTC. NTP also supports a negative leap second, where a second is
removed from UTC if it ever becomes necessary.

While earlier versions of NTP contained some ambiguity regarding when
a leap second broadcast by a server should be applied by a client,
RFC 5905 is clear that leap seconds are only applied on the last day
of a month. However, because some older clients may apply it at the
end of the current day, it is RECOMMENDED that NTP servers wait until
the last day of the month before broadcasting leap seconds. Doing
this will prevent older clients from applying a leap second at the
wrong time. When implementing this recommendation, operators should
ensure that clients are not configured to use polling intervals
greater than 24 hours so the leap-second notification is not missed.

In circumstances where an NTP server is not receiving leap-second
information from an automated source, certain organizations maintain
files that are updated every time a new leap second is announced:

    NIST: <ftp://time.nist.gov/pub/leap-seconds.list>

    US Navy (maintains GPS Time):
    <ftp://tycho.usno.navy.mil/pub/ntp/leap-seconds.list>

    IERS (announces leap seconds):
    <https://hpiers.obspm.fr/iers/bul/bulc/ntp/leap-seconds.list>

3.7.1. Leap Smearing

Some NTP installations make use of a technique called leap smearing.
With this method, instead of introducing an extra second (or
eliminating a second) in a leap-second event, NTP time is adjusted in
small increments over a comparably large window of time (called the
smear interval) around the leap-second event. The smear interval

should be large enough for the time to be adjusted at a low rate, so
that clients will follow the smeared time without objecting. Periods
ranging from two to twenty-four hours have been used successfully.
During the adjustment window, all the NTP clients' times may be
offset from UTC by as much as a full second, depending on the
implementation. However, all clients will generally agree on what
time they think it is.

The purpose of leap smearing is to enable systems that don't deal
with the leap-second event properly to function consistently, at the
expense of fidelity to UTC during the smear window. During a
standard leap-second event, a minute will have 61 (or possibly 59)
seconds, and some applications (and even some OSs) are known to have
problems with that.

Operators who have legal obligations or other strong requirements to
be synchronized with UTC or civil time SHOULD NOT use leap smearing
because the distributed time cannot be guaranteed to be traceable to
UTC during the smear interval.

Clients that are connected to leap-smearing servers MUST NOT apply
the standard NTP leap-second handling. These clients must never have
a leap-second file loaded, and the smearing servers must never
advertise to clients for which a leap second is pending.

Any use of leap-smearing servers should be limited to within a
single, well-controlled environment. Leap smearing MUST NOT be used
for public-facing NTP servers, as they will disagree with non-
smearing servers (as well as UTC) during the leap smear interval, and
there is no standardized way for a client to detect that a server is
using leap smearing. However, be aware that some public-facing
servers may be configured this way in spite of this guidance.

System administrators are advised to be aware of impending leap
seconds and how the servers (inside and outside their organization)
they are using deal with them. Individual clients MUST NOT be
configured to use a mixture of smeared and non-smeared servers. If a
client uses smeared servers, the servers it uses must all have the
same leap smear configuration.

4. NTP Security Mechanisms

In the standard configuration, NTP packets are exchanged unprotected
between client and server. An adversary that is able to become a man
in the middle is therefore able to drop, replay, or modify the
content of the NTP packet, which leads to degradation of the time
synchronization or transmission of false time information. A threat
analysis for time-synchronization protocols is given in [RFC7384].

NTP provides two internal security mechanisms to protect the
authenticity and integrity of the NTP packets. Both measures protect
the NTP packet by means of a Message Authentication Code (MAC).
Neither of them encrypts the NTP's payload because this payload
information is not considered to be confidential.

4.1. Pre-Shared Key Approach

This approach applies a symmetric key for the calculation of the MAC,
which protects the authenticity and integrity of the exchanged
packets for an association. NTP does not provide a mechanism for the
exchange of keys between the associated nodes. Therefore, for each
association, keys MUST be exchanged securely by external means, and
they MUST be protected from disclosure. It is RECOMMENDED that each
association be protected by its own unique key. It is RECOMMENDED
that participants agree to refresh keys periodically. However, NTP
does not provide a mechanism to assist in doing so. Each
communication partner will need to keep track of its keys in its own
local key storage.

[RFC5905] specifies using the MD5 hash algorithm for calculation of
the MAC, but other algorithms may be supported as well. The MD5 hash
is now considered to be too weak and unsuitable for cryptographic
usage. [RFC6151] has more information on the algorithm's weaknesses.
Implementations will soon be available based on AES-128-CMAC
[RFC8573], and users SHOULD use that when it is available.

Some implementations store the key in clear text. Therefore, it MUST
only be readable by the NTP process.

An NTP client has to be able to link a key to a particular server in
order to establish a protected association. This linkage is
implementation specific. Once applied, a key will be trusted until
the link is removed.

4.2. Autokey

[RFC5906] specifies the Autokey protocol. It was published in 2010
to provide automated key management and authentication of NTP
servers. However, security researchers have identified
vulnerabilities [AUTOKEY] in the Autokey protocol.

Autokey SHOULD NOT be used.

4.3. Network Time Security

Work is in progress on an enhanced replacement for Autokey. Refer to
[NTSFORNTP] for more information.

4.4. External Security Protocols

   If applicable, external security protocols such as IPsec and MACsec
   can be applied to enhance the integrity and authenticity protection
   of NTP time-synchronization packets. Usage of such external security
   protocols can decrease time-synchronization performance [RFC7384].
   Therefore, operators are advised to carefully evaluate whether the
   decreased time-synchronization performance meets their specific
   timing requirements.

   Note that none of the security measures described in Section 4 can
   prevent packet delay manipulation attacks on NTP. Such delay attacks
   can target time-synchronization packets sent as clear text or even
   within an encrypted tunnel. These attacks are described further in
   Section 3.2.6 of [RFC7384].

5. NTP Security Best Practices

   This section lists some general NTP security practices, but these
   issues may (or may not) have been mitigated in particular versions of
   particular implementations. Contact the maintainers of the relevant
   implementation for more information.

5.1. Minimizing Information Leakage

   The base NTP packet leaks important information (including reference
   ID and reference time) that may be used in attacks [NDSS16]
   [CVE-2015-8138] [CVE-2016-1548]. A remote attacker can learn this
   information by sending mode 3 queries to a target system and
   inspecting the fields in the mode 4 response packet. NTP control
   queries also leak important information (including reference ID,
   expected origin timestamp, etc.) that may be used in attacks
   [CVE-2015-8139]. A remote attacker can learn this information by
   sending control queries to a target system and inspecting the leaked
   information in the response.

   As such, mechanisms outside of the NTP protocol, such as Access
   Control Lists, SHOULD be used to limit the exposure of this
   information to allowed IP addresses and keep it from remote attackers
   not on the list. Hosts SHOULD only respond to NTP control queries
   from authorized parties.

   An NTP client that does not provide time on the network can
   additionally log and drop incoming mode 3 timing queries from
   unexpected sources. Note well that the easiest way to monitor the
   status of an NTP instance is to send it a mode 3 query, so it may not
   be desirable to drop all mode 3 queries. As an alternative,
   operators SHOULD either filter mode 3 queries from outside their

networks or make sure mode 3 queries are allowed only from trusted
systems or networks.

A "leaf-node host" is a host that uses NTP solely for the purpose of
adjusting its own system time. Such a host is not expected to
provide time to other hosts and relies exclusively on NTP's basic
mode to take time from a set of servers (that is, the host sends mode
3 queries to its servers and receives mode 4 responses from these
servers containing timing information.) To minimize information
leakage, leaf-node hosts SHOULD drop all incoming NTP packets except
mode 4 response packets that come from known sources. An exception
to this can be made if a leaf-node host is being actively monitored,
in which case incoming packets from the monitoring server can be
allowed.

Please refer to [DATAMIN] for more information.

5.2. Avoiding Daemon Restart Attacks

   [RFC5905] says NTP clients should not accept time shifts greater than
   the panic threshold. Specifically, RFC 5905 says "PANIC means the
   offset is greater than the panic threshold PANICT (1000 s) and SHOULD
   cause the program to exit with a diagnostic message to the system
   log."

   However, this behavior can be exploited by attackers as described in
   [NDSS16] when the following two conditions hold:

   1. The operating system automatically restarts the NTP client when
       it quits. Modern UNIX and UNIX-like operating systems are
       replacing traditional init systems with process supervisors, such
       as systemd, which can be configured to automatically restart any
       daemons that quit. This behavior is the default in CoreOS and
       Arch Linux. As of the time of this writing, it appears likely to
       become the default behavior in other systems as they migrate
       legacy init scripts to process supervisors such as systemd.

   2. The NTP client is configured to ignore the panic threshold on all
       restarts.

   In such cases, if the attacker can send the target an offset that
   exceeds the panic threshold, the client will quit. Then, when it
   restarts, it ignores the panic threshold and accepts the attacker's
   large offset.

   Operators need to be aware that when operating with the above two
   conditions, the panic threshold offers no protection from attacks.
   The natural solution is not to run hosts with these conditions.

Specifically, operators SHOULD NOT ignore the panic threshold in all
cold-start situations unless sufficient oversight and checking is in
place to make sure that this type of attack cannot happen.

As an alternative, the following steps MAY be taken by operators to
mitigate the risk of attack:

o Monitor the NTP system log to detect when the NTP daemon quit due
   to a panic event, as this could be a sign of an attack.

o Request manual intervention when a timestep larger than the panic
   threshold is detected.

o Configure the ntp client to only ignore the panic threshold in a
   cold-start situation.

o Increase the minimum number of servers required before the NTP
   client adjusts the system clock. This will make the NTP client
   wait until enough trusted sources of time agree before declaring
   the time to be correct.

In addition, the following steps SHOULD be taken by those who
implement the NTP protocol:

o Prevent the NTP daemon from taking time steps that set the clock
   to a time earlier than the compile date of the NTP daemon.

o Prevent the NTP daemon from putting 'INIT' in the reference ID of
   its NTP packets upon initializing. This will make it more
   difficult for attackers to know when the daemon reboots.

5.3. Detection of Attacks through Monitoring

   Operators SHOULD monitor their NTP instances to detect attacks. Many
   known attacks on NTP have particular signatures. Common attack
   signatures include:

   1. Bogus packets - A packet whose origin timestamp does not match
       the value that is expected by the client.

   2. Zero origin packet - A packet with an origin timestamp set to
       zero [CVE-2015-8138].

   3. A packet with an invalid cryptographic MAC.

   The observation of many such packets could indicate that the client
   is under attack.

5.4. Kiss-o'-Death Packets

   The "Kiss-o'-Death" (KoD) packet includes a rate-management mechanism
   where a server can tell a misbehaving client to reduce its query
   rate. KoD packets in general (and the RATE packet in particular) are
   defined in Section 7.4 of [RFC5905]. It is RECOMMENDED that all NTP
   devices respect these packets and back off when asked to do so by a
   server. This is even more important for an embedded device, which
   may not have an exposed control interface for NTP.

   That said, a client MUST only accept a KoD packet if it has a valid
   origin timestamp. Once a RATE packet is accepted, the client should
   increase its poll interval value (thus decreasing its polling rate)
   to a reasonable maximum. This maximum can vary by implementation but
   should not exceed a poll interval value of 13 (two hours). The
   mechanism to determine how much to increase the poll interval value
   is undefined in [RFC5905]. If the client uses the poll interval
   value sent by the server in the RATE packet, it MUST NOT simply
   accept any value. Using large interval values may create a vector
   for a denial-of-service attack that causes the client to stop
   querying its server [NDSS16].

   The KoD rate-management mechanism relies on clients behaving properly
   in order to be effective. Some clients ignore the RATE packet
   entirely, and other poorly implemented clients might unintentionally
   increase their poll rate and simulate a denial-of-service attack.
   Server administrators are advised to be prepared for this and take
   measures outside of the NTP protocol to drop packets from misbehaving
   clients when these clients are detected.

   Kiss-o'-Death (KoD) packets can be used in denial-of-service attacks.
   Thus, the observation of even just one RATE packet with a high poll
   value could be sign that the client is under attack. And KoD packets
   are commonly accepted even when not cryptographically authenticated,
   which increases the risk of denial-of-service attacks.

5.5. Broadcast Mode Only on Trusted Networks

   Per [RFC5905], NTP's broadcast mode is authenticated using symmetric
   key cryptography. The broadcast server and all its broadcast clients
   share a symmetric cryptographic key, and the broadcast server uses
   this key to append a MAC to the broadcast packets it sends.

   Importantly, all broadcast clients that listen to this server have to
   know the cryptographic key. This means that any client can use this
   key to send valid broadcast messages that look like they come from
   the broadcast server. Thus, a rogue broadcast client can use its
   knowledge of this key to attack the other broadcast clients.

For this reason, an NTP broadcast server and all its clients have to
trust each other. Broadcast mode SHOULD only be run from within a
trusted network.

5.6. Symmetric Mode Only with Trusted Peers

In symmetric mode, two peers, Alice and Bob, can both push and pull
synchronization to and from each other using either ephemeral
symmetric passive (mode 2) or persistent symmetric active (NTP mode
1) packets. The persistent association is preconfigured and
initiated at the active peer but is not preconfigured at the passive
peer (Bob). Upon receipt of a mode 1 NTP packet from Alice, Bob
mobilizes a new ephemeral association if he does not have one
already. This is a security risk for Bob because an arbitrary
attacker can attempt to change Bob's time by asking Bob to become its
symmetric passive peer.

For this reason, a host SHOULD only allow symmetric passive
associations to be established with trusted peers. Specifically, a
host SHOULD require each of its symmetric passive associations to be
cryptographically authenticated. Each symmetric passive association
SHOULD be authenticated under a different cryptographic key.

6. NTP in Embedded Devices

As computing becomes more ubiquitous, there will be many small
embedded devices that require accurate time. These devices may not
have a persistent battery-backed clock, so using NTP to set the
correct time on power-up may be critical for proper operation. These
devices may not have a traditional user interface, but if they
connect to the Internet, they will be subject to the same security
threats as traditional deployments.

6.1. Updating Embedded Devices

Vendors of embedded devices are advised to pay attention to the
current state of protocol security issues and bugs in their chosen
implementation because their customers don't have the ability to
update their NTP implementation on their own. Those devices may have
a single firmware upgrade, provided by the manufacturer, that updates
all capabilities at once. This means that the vendor assumes the
responsibility of making sure their devices have an up-to-date and
secure NTP implementation.

Vendors of embedded devices SHOULD include the ability to update the
list of NTP servers used by the device.

There is a catalog of NTP server abuse incidents, some of which
involve embedded devices, on the Wikipedia page for NTP Server Misuse
and Abuse [MISUSE].

6.2. Server Configuration

Vendors of embedded devices with preconfigured NTP servers need to
carefully consider which servers to use. There are several public-
facing NTP servers available, but they may not be prepared to service
requests from thousands of new devices on the Internet. Vendors MUST
only preconfigure NTP servers that they have permission to use.

Vendors are encouraged to invest resources into providing their own
time servers for their devices to connect to. This may be done
through the NTP Pool Project, as documented in Section 3.6.

Vendors should read [RFC4085], which advises against embedding
globally routable IP addresses in products and offers several better
alternatives.

7. NTP over Anycast

Anycast is described in BCP 126 [RFC4786] (see also [RFC7094]). With
anycast, a single IP address is assigned to multiple servers, and
routers direct packets to the closest active server.

Anycast is often used for Internet services at known IP addresses,
such as DNS. Anycast can also be used in large organizations to
simplify the configuration of many NTP clients. Each client can be
configured with the same NTP server IP address, and a pool of anycast
servers can be deployed to service those requests. New servers can
be added to or taken from the pool, and other than a temporary loss
of service while a server is taken down, these additions can be
transparent to the clients.

Note well that using a single anycast address for NTP presents its
own potential issues. It means each client will likely use a single
time server source. A key element of a robust NTP deployment is each
client using multiple sources of time. With multiple time sources, a
client will analyze the various time sources, select good ones, and
disregard poor ones. If a single anycast address is used, this
analysis will not happen. This can be mitigated by creating
multiple, separate anycast pools so clients can have multiple sources
of time while still gaining the configuration benefits of the anycast
pools.

If clients are connected to an NTP server via anycast, the client
does not know which particular server they are connected to. As
anycast servers enter and leave the network or the network topology
changes, the server to which a particular client is connected may
change. This may cause a small shift in time from the perspective of
the client when the server to which it is connected changes. Extreme
cases where the network topology changes rapidly could cause the
server seen by a client to rapidly change as well, which can lead to
larger time inaccuracies. It is RECOMMENDED that network operators
only deploy anycast NTP in environments where operators know these
small shifts can be tolerated by the applications running on the
clients being synchronized in this manner.

Configuration of an anycast interface is independent of NTP. Clients
will always connect to the closest server, even if that server is
having NTP issues. It is RECOMMENDED that anycast NTP
implementations have an independent method of monitoring the
performance of NTP on a server. If the server is not performing to
specification, it should remove itself from the anycast network. It
is also RECOMMENDED that each anycast NTP server have an alternative
method of access, such as an alternate unicast IP address, so its
performance can be checked independently of the anycast routing
scheme.

One useful application in large networks is to use a hybrid unicast/
anycast approach. Stratum 1 NTP servers can be deployed with unicast
interfaces at several sites. Each site may have several Stratum 2
servers with two Ethernet interfaces or a single interface that can
support multiple addresses. One interface has a unique unicast IP
address. The second has an anycast IP interface (with a shared IP
address per location). The unicast interfaces can be used to obtain
time from the Stratum 1 servers globally (and perhaps peer with the
other Stratum 2 servers at their site). Clients at each site can be
configured to use the shared anycast address for their site,
simplifying their configuration. Keeping the anycast routing
restricted on a per-site basis will minimize the disruption at the
client if its closest anycast server changes. Each Stratum 2 server
can be uniquely identified on their unicast interface to make
monitoring easier.

8. IANA Considerations

This document has no IANA actions.

9. Security Considerations

   Time is a fundamental component of security on the Internet. The
   absence of a reliable source of current time subverts many common web
   authentication schemes, e.g., by allowing the use of expired
   credentials or allowing the replay of messages only intended to be
   processed once.

   Much of this document directly addresses how to secure NTP servers.
   In particular, see Section 2, Section 4, and Section 5.

   There are several general threats to time-synchronization protocols,
   which are discussed in [RFC7384].

   [NTSFORNTP] specifies the Network Time Security (NTS) mechanism and
   applies it to NTP. Readers are encouraged to check the status of the
   document and make use of the methods it describes.

10. References

10.1. 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>.

   [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering:
              Defeating Denial of Service Attacks which employ IP Source
              Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827,
              May 2000, <https://www.rfc-editor.org/info/rfc2827>.

   [RFC4085] Plonka, D., "Embedding Globally-Routable Internet
              Addresses Considered Harmful", BCP 105, RFC 4085,
              DOI 10.17487/RFC4085, June 2005,
              <https://www.rfc-editor.org/info/rfc4085>.

   [RFC4786] Abley, J. and K. Lindqvist, "Operation of Anycast
              Services", BCP 126, RFC 4786, DOI 10.17487/RFC4786,
              December 2006, <https://www.rfc-editor.org/info/rfc4786>.

   [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>.

   [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>.

10.2. Informative References

   [AUTOKEY] Roettger, S., "Autokey-Protocol Analysis", August 2011,
              <https://lists.ntp.org/pipermail/
              ntpwg/2011-August/001714.html>.

   [BCP38WIKI]
              "BCP38.info Wiki", October 2016, <http://www.bcp38.info>.

   [CCR16]   Malhotra, A. and S. Goldberg, "Attacking NTP's
              Authenticated Broadcast Mode", SIGCOMM Computer
              Communications Review (CCR) Volume 46, Issue 2,
              DOI 10.1145/2935634.2935637, April 2016.

   [CONFIGNTP]
              Network Time Foundation, "Configuring NTP", November 2018,
              <https://support.ntp.org/bin/view/Support/ConfiguringNTP>.

   [CTRLMSG] Haberman, B., Ed., "Control Messages Protocol for Use with
              Network Time Protocol Version 4", Work in Progress,
              draft-ietf-ntp-mode-6-cmds-06, September 2018.

   [CVE-2015-8138]
              Van Gundy, M. and J. Gardner, "Network Time Protocol
              Origin Timestamp Check Impersonation Vulnerability",
              January 2016,
              <https://www.talosintel.com/reports/TALOS-2016-0077>.

   [CVE-2015-8139]
              Van Gundy, M., "Network Time Protocol ntpq and ntpdc
              Origin Timestamp Disclosure Vulnerability", January 2016,
              <https://www.talosintel.com/reports/TALOS-2016-0078>.

   [CVE-2016-1548]
              Gardner, J. and M. Lichvar, "Xleave Pivot: NTP Basic Mode
              to Interleaved", April 2016,
              <https://blog.talosintel.com/2016/04/
              vulnerability-spotlight-further-ntpd_27.html>.

   [DATAMIN] Franke, D. and A. Malhotra, "NTP Client Data
              Minimization", Work in Progress, draft-ietf-ntp-data-
              minimization-04, March 2019.

   [DDOS]   Prince, M., "Technical Details Behind a 400Gbps NTP
               Amplification DDoS Attack", February 2014,
               <https://blog.cloudflare.com/technical-details-behind-a-
               400gbps-ntp-amplification-ddos-attack>.

   [IERS]   IERS, "IERS Bulletins",
               <https://www.iers.org/IERS/EN/Publications/Bulletins/
               bulletins.html>.

   [IMC14]   Czyz, J., Kallitsis, M., Gharaibeh, M., Papadopoulos, C.,
               Bailey, M., and M. Karir, "Taming the 800 Pound Gorilla:
               The Rise and Decline of NTP DDoS Attacks", Proceedings of
               the 2014 Internet Measurement Conference,
               DOI 10.1145/2663716.2663717, November 2014.

   [LEAPSEC]   Burnicki, M., "Leap Second Smearing", <http://bk1.ntp.org/
               ntp-stable/README.leapsmear?PAGE=anno>.

   [MILLS2006]
               Mills, D., "Computer network time synchronization: the
               Network Time Protocol", CRC Press, 2006.

   [MISUSE]   Wikipedia, "NTP Server Misuse and Abuse", May 2019,
               <https://en.wikipedia.org/w/index.php?
               title=NTP_server_misuse_and_abuse&oldid=899024751>.

   [NDSS14]   Rossow, C., "Amplification Hell: Revisiting Network
               Protocols for DDoS Abuse", Network and Distributed System
               Security (NDSS) Symposium 2014,
               DOI 10.14722/ndss.2014.23233, February 2014,
               <https://www.ndss-symposium.org/ndss2014/programme/
               amplification-hell-revisiting-network-protocols-ddos-
               abuse/>.

   [NDSS16]   Malhotra, A., Cohen, I., Brakke, E., and S. Goldberg,
               "Attacking the Network Time Protocol", Network and
               Distributed System Security (NDSS) Symposium 2016,
               DOI 10.14722/ndss.2016.23090, February 2016,
               <https://eprint.iacr.org/2015/1020.pdf>.

   [NTPDOWN]   Network Time Foundation, "NTP Software Downloads",
               <http://www.ntp.org/downloads.html>.

   [NTSFORNTP]
               Franke, D., Sibold, D., Teichel, K., Dansarie, M., and R.
               Sundblad, "Network Time Security for the Network Time
               Protocol", Work in Progress, draft-ietf-ntp-using-nts-for-
               ntp-20, July 2019.

   [POOLUSE] NTP Pool Project, "Use the Pool",
              <https://www.pool.ntp.org/en/use.html>.

   [POOLVENDOR]
              NTP Pool Project, "The NTP Pool for Vendors",
              <https://www.pool.ntp.org/en/vendors.html>.

   [RFC1305] Mills, D., "Network Time Protocol (Version 3)
              Specification, Implementation and Analysis", RFC 1305,
              DOI 10.17487/RFC1305, March 1992,
              <https://www.rfc-editor.org/info/rfc1305>.

   [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>.

   [RFC6151] Turner, S. and L. Chen, "Updated Security Considerations
              for the MD5 Message-Digest and the HMAC-MD5 Algorithms",
              RFC 6151, DOI 10.17487/RFC6151, March 2011,
              <https://www.rfc-editor.org/info/rfc6151>.

   [RFC7094] McPherson, D., Oran, D., Thaler, D., and E. Osterweil,
              "Architectural Considerations of IP Anycast", RFC 7094,
              DOI 10.17487/RFC7094, January 2014,
              <https://www.rfc-editor.org/info/rfc7094>.

   [RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in
              Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384,
              October 2014, <https://www.rfc-editor.org/info/rfc7384>.

   [RFC8573] Malhotra, A. and S. Goldberg, "Message Authentication Code
              for the Network Time Protocol", RFC 8573,
              DOI 10.17487/RFC8573, June 2019,
              <https://www.rfc-editor.org/info/rfc8573>.

   [SOLAR]   Wikipedia, "Solar Time", May 2019,
              <https://en.wikipedia.org/w/index.php?
              title=Solar_time&oldid=896601472#Mean_solar_time>.

Appendix A. Best Practices Specific to the Network Time Foundation
             Implementation

   The Network Time Foundation (NTF) provides a widely used
   implementation of NTP, known as ntpd [NTPDOWN]. It is an evolution
   of the first NTP implementations developed by David Mills at the
   University of Delaware. This appendix contains additional
   recommendations specific to this implementation that are valid at the
   time of this writing.

A.1. Use Enough Time Sources

   In addition to the recommendation given in Section 3.2, the ntpd
   implementation provides the 'pool' directive. Starting with ntp-
   4.2.6, using this directive in the ntp.conf file will spin up enough
   associations to provide robust time service and will disconnect poor
   servers and add in new servers as needed. The 'minclock' and
   'maxclock' options of the 'tos' command may be used to override the
   default values of how many servers are discovered through the 'pool'
   directive.

A.2. NTP Control and Facility Messages

   In addition to NTP control messages, the ntpd implementation also
   offers the Mode 7 commands for monitoring and configuration.

   If Mode 7 has been explicitly enabled to be used for more than basic
   monitoring, it should be limited to authenticated sessions that
   provide a 'requestkey'.

   As mentioned above, there are two general ways to use Mode 6 and Mode
   7 requests. One way is to query ntpd for information, and this mode
   can be disabled with:

   restrict ... noquery

   The second way to use Mode 6 and Mode 7 requests is to modify ntpd's
   behavior. Modification of ntpd's configuration requires an
   authenticated session by default. If no authentication keys have
   been specified, no modifications can be made. For additional
   protection, the ability to perform these modifications can be
   controlled with:

   restrict ... nomodify

   Users can prevent their NTP servers from considering query/
   configuration traffic by default by adding the following to their
   ntp.conf file:

   restrict default -4 nomodify notrap nopeer noquery

   restrict default -6 nomodify notrap nopeer noquery

   restrict source nomodify notrap noquery

A.3. Monitoring

   The ntpd implementation allows remote monitoring. Access to this
   service is generally controlled by the "noquery" directive in NTP's
   configuration file (ntp.conf) via a "restrict" statement. The syntax
   reads:

   restrict address mask address_mask noquery

   If a system is using broadcast mode and is running ntp-4.2.8p6 or
   later, use the fourth field of the ntp.keys file to specify the IPs
   of machines that are allowed to serve time to the group.

A.4. Leap-Second File

   The use of leap-second files requires ntpd 4.2.6 or later. After
   fetching the leap-second file onto the server, add this line to
   ntpd.conf to apply and use the file, substituting the proper path:

   leapfile "/path/to/leap-file"

   There may be a need to restart ntpd to apply this change.

   ntpd servers with a manually configured leap-second file will ignore
   a leap-second information broadcast from upstream NTP servers until
   the leap-second file expires. If no valid leap-second file is
   available, then a leap-second notification from an attached reference
   clock is always accepted by ntpd.

   If no valid leap-second file is available, a leap-second notification
   may be accepted from upstream NTP servers. As of ntp-4.2.6, a
   majority of servers must provide the notification before it is
   accepted. Before 4.2.6, a leap-second notification would be accepted
   if a single upstream server of a group of configured servers provided
   a leap-second notification. This would lead to misbehavior if single
   NTP servers sent an invalid leap second warning, e.g., due to a
   faulty GPS receiver in one server, but this behavior was once chosen
   because in the "early days", there was a greater chance that leap-

   second information would be available from a very limited number of
   sources.

A.5. Leap Smearing

   Leap smearing was introduced in ntpd versions 4.2.8.p3 and 4.3.47 in
   response to client requests. Support for leap smearing is not
   configured by default and must be added at compile time. In
   addition, no leap smearing will occur unless a leap smear interval is
   specified in ntpd.conf. For more information, refer to [LEAPSEC].

A.6. Configuring ntpd

   See [CONFIGNTP] for additional information on configuring ntpd.

A.7. Pre-Shared Keys

   Each communication partner must add the key information to their key
   file in the form:

   keyid type key

   where "keyid" is a number between 1 and 65534, inclusive; "type" is
   an ASCII character that defines the key format; and "key" is the key
   itself.

   An ntpd client establishes a protected association by appending the
   option "key keyid" to the server statement in ntp.conf,

   server address key keyid

   substituting the server address in the "address" field and the
   numerical keyid to use with that server in the "keyid" field.

   A key is deemed trusted when its keyid is added to the list of
   trusted keys by the "trustedkey" statement in ntp.conf.

   trustedkey keyid_1 keyid_2 ... keyid_n

   Starting with ntp-4.2.8p7, the ntp.keys file accepts an optional
   fourth column, a comma-separated list of IPs that are allowed to
   serve time. Use this feature. Note, however, that an adversarial
   client that knows the symmetric broadcast key could still easily
   spoof its source IP to an IP that is allowed to serve time. This is
   easy to do because the origin timestamp on broadcast mode packets is
   not validated by the client. By contrast, client/server and
   symmetric modes do require origin timestamp validation, making it
   more difficult to spoof packets [CCR16].

Authors' Addresses

   Denis Reilly (editor)
   Orolia USA
   1565 Jefferson Road, Suite 460
   Rochester, NY 14623
   United States of America

   Email: denis.reilly@orolia.com

   Harlan Stenn
   Network Time Foundation
   P.O. Box 918
   Talent, OR 97540
   United States of America

   Email: stenn@nwtime.org

   Dieter Sibold
   Physikalisch-Technische Bundesanstalt
   Bundesallee 100
   Braunschweig D-38116
   Germany

   Phone: +49-(0)531-592-8420
   Fax: +49-531-592-698420
   Email: dieter.sibold@ptb.de