Workgroup: Network Working Group
Internet-Draft: draft-ietf-ntp-chronos-12
Published: 7 November 2022
Intended Status: Informational
Expires: 11 May 2023
Authors: N. Rozen-Schiff
Hebrew University of Jerusalem
D. Dolev
Hebrew University of Jerusalem
T. Mizrahi
Huawei Network.IO Innovation Lab
M. Schapira
Hebrew University of Jerusalem

A Secure Selection and Filtering Mechanism for the Network Time Protocol with Khronos

Abstract

The Network Time Protocol version 4 (NTPv4), as defined in RFC 5905, is the mechanism used by NTP clients to synchronize with NTP servers across the Internet. This document specifies an extension to the NTPv4 client, named Khronos, which is used as a "watchdog" alongside NTPv4, and provides improved security against time shifting attacks. Khronos involves changes to the NTP client's system process only. Since it does not affect the wire protocol, the Khronos mechanism is applicable to any current or future time protocol.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

NTPv4, as defined in RFC 5905 [RFC5905], is vulnerable to time shifting attacks, in which the attacker changes (shifts) the clock of a network device. Time shifting attacks on NTP clients can be based on interfering the communication between the NTP clients and servers or compromising the servers themselves. Time shifting attacks on NTP are possible even if NTP communication is encrypted and authenticated. A weaker man-in-the-middle (MitM) attacker can shift time simply by dropping or delaying packets, whereas a powerful attacker, who has full control over an NTP server, can do so by explicitly determining the NTP response content. This document introduces a time shifting mitigation mechanism called Khronos. Khronos can be integrated into NTPv4-compatible servers as an NTPv4 client's "watchdog" against time shifting attacks. An NTP client that runs Khronos is interoperable with [RFC5905]-compatible NTPv4 servers. The Khronos mechanism does not affect the wire mechanism and is therefore applicable to any current or future time protocol.

Khronos is a mechanism that runs in the background, continuously monitoring client clock (which is updated by NTPv4) and calculating an estimated offset which we refer by "Khronos time offset". When the offset exceeds a predefined threshold (specified in Section 4.2), this is interpreted as the client experiencing a time shifting attack. In this case, Khronos updates the client's clock.

When the client is not under attack, Khronos is passive, allowing NTPv4 to control the client's clock and providing the ordinary high precision and accuracy of NTPv4. When under attack, Khronos takes control over the client's clock, mitigating the time shift, while guaranteeing relatively high accuracy with respect to UTC and precision, as discussed in Section 6.

By leveraging techniques from distributed computing theory for time-synchronization in the presence of Byzantine attackers, Khronos achieves accurate synchronization even in the presence of powerful attackers who are in direct control of a large number of NTP servers. Khronos will prevent shifting the clock when the ratio of compromised time samples is below 2/3. In each polling interval, Khorons client randomly selects and samples a few NTP servers out of a local pool of hundreds of servers. Khronos is carefully engineered to minimize the load on NTP servers and the communication overhead. In contrast, NTPv4, employs an algorithm which typically relies on a small subset of the NTP server pool (e.g., 4 servers) for time synchronization, and is much more vulnerable to time shifting attacks. Configuring NTPv4 to use several hundreds of servers will increase its security, but will incur very high network and computational overhead compared to Khronos and will be bounded by compromised ratio of half of the time samples.

A Khronos client iteratively "crowdsources" time queries across NTP servers and applies a provably secure algorithm for eliminating "suspicious" responses and for averaging over the remaining responses. In each Khronos poll interval, the Khronos client selects, uniformly at random, a small subset (e.g., 10-15 servers) of a large server pool (containing hundreds of servers). To minimize the load on NTP servers and the communication overhead, the frequency of Khronos poll intervals should be much less dense than that of standard NTPv4 clock updates (e.g., the Khronos clock can be updated once every 10 NTPv4 clock updates). Khronos' security was evaluated both theoretically and experimentally with a prototype implementation. According to this security analyses, if a local Khronos pool consists of, for example, 500 servers, 1/7 of whom are controlled by a man-in-the-middle, attacker and Khronos queries 15 servers in each Khronos poll interval (around 10 times the NTPv4 poll interval), then over 20 years of effort are required (in expectation) to successfully shift time at a Khronos client by over 100 ms from UTC. The full exposition of the formal analysis of this guarantee is available at [Khronos_paper].

Khronos introduces a watchdog mechanism that maintains a time offset value that is used as a reference for detecting attacks. The time offset value computation differs from the current NTPv4 in two key aspects. First, Khronos periodically synchronizes, in each Khronos poll interval, with only a few (tens) randomly selected servers out of a pool consisting of a large number (e.g., hundreds) of NTP servers, thereby providing high security while minimizing the load on the NTP servers. Second, Khronos computes "Khronos time offset" based on an approximate agreement technique to remove outliers, thus limiting the attacker's ability to contaminate the "time samples" (offsets) derived from the queried NTP servers. These two elements of Khronos' design provide provable security guarantees against both man-in-the-middle attackers and attackers capable of compromising a large number of NTP servers.

2. Conventions Used in This Document

2.1. Terminology

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 [RFC2119].

2.2. Terms and Abbreviations

NTPv4 Network Time Protocol version 4 [RFC5905].

System process Selection Algorithm and the Cluster Algorithm [RFC5905].

Security Requirements Security Requirements of Time Protocols in Packet Switched Networks [RFC7384].

NTS Network Time Security for the Network Time Protocol [RFC8915].

2.3. Notations

Describing Khronos algorithm, the following notation is used.

Notation	Meaning
n	The number of candidate servers in Khronos pool (potentially hundreds).
m	

Notation	Meaning
	The number of servers that Khronos queries in each poll interval (up to tens).
w	An upper bound on the distance between any "truechimer" NTP server (as in [RFC5905]) and UTC.
B	An upper bound on the client's clock error rate (ms/sec).
ERR	An upper bound on the client's clock error between Khronos polls (ms).
K	The number of Khronos pool re-samplings until reaching "Panic mode".
H	Predefined threshold for time offset triggering clock update by Khronos.

Table 1: Khronos Notations

The recommended values are discussed in Section 3.3.

3. Khronos' Design

Khronos watchdog periodically queries a set of m (tens) servers from a large (hundreds) server pool in each Khronos poll interval, where the m servers are selected from the server pool at random. Based on empirical analyses, to minimize the load on NTP servers while providing high security, the Khronos poll interval should be around 10 times the NTPv4 poll interval (i.e., a Khronos clock update occurs once every 10 NTPv4 clock updates). In each Khronos poll interval, if the Khronos time offset exceeds a predetermined threshold (denoted as H), an attack is indicated.

Unless an attack is indicated, Khronos uses only one sample from each server (avoiding "Clock Filter Algorithm" as defined in Section 10 in [RFC5905]). When under attack, Khronos uses several samples from each server, and executes the "Clock Filter Algorithm" for choosing the best sample from each server, with low jitter. Then, given a sample from each server, Khronos discards outliers by executing the procedure described in Section 3.2.

Between consecutive Khronos polls, Khronos keeps track of clock offsets, for example by catching clock discipline (as in [RFC5905]) calls. The sum of offsets is referred as "Khronos inter-poll offset" (denoted as tk) which is set to zero after each Khronos poll.

3.1. Khronos Calibration

Calibration is performed at the first time the Khronos is executed, and also periodically, once in a long time (e.g., every few weeks/ months). The calibration process generates a local Khronos pool of n (up to hundreds) NTP servers the client can synchronize with. To this end, Khronos makes DNS queries to addresses of NTP pools collect the union of all received IP addresses. The servers in the

Khronos pool should be scattered across different regions to make it harder for an attacker to compromise, or gain man-in-the-middle capabilities, with respect to a large fraction of the Khronos pool. Therefore, Khronos calibration queries general NTP server pools (for example pool.org), and not only the pool in the client's state or region.

The first Khronos update requires m servers, which can be found in several minutes. Moreover, it is possible to query several DNS pool names (for example 0.pool.ntp.org, 1.pool.ntp.org etc. and regional pools) to vastly accelerate the calibration and the first update. By storing the retrieved addresses to permanent storage a recalibration will be avoided in case of system restart.

The calibration is the only Khronos part where DNS traffic is generated. Around 250 DNS queries are required by Khronos to obtain a pool of 1000 NTP servers. Assuming the calibration period is one month, the expected DNS traffic generated by Khorons client is less than 10 DNS queries per day, which is usually several orders of magnitude lower than the total daily number of DNS queries per machine.

3.2. Khronos' Poll and System Processes

In each Khronos poll interval the Khronos system process randomly chooses a set of m (tens) servers out of the Khronos pool of n (hundreds) servers. Khronos server polling times can be spread uniformly, similar to NTPv4. Servers which do not respond during the Khronos poll are filtered out. If less than 1/3 of the m servers are left, a new subset of servers is immediately sampled, in the exact same manner (called "resampling" process).

Next, out of the time-samples received from this chosen subset of servers, the lowest third of the samples' offset values and highest third of the samples' offset values are discarded.

Khronos checks that the following two conditions hold for the remaining sampled offsets:

* The maximal distance between every two offsets does not exceed 2w (can be verified by considering just the minimum and the maximum offsets).

* The distance between the offsets average and Khronos inter-poll offset is at most ERR+2w.

(where w and ERR are as described in Table 1).

In the event that both of these conditions are satisfied, the average of the offsets is set to be the "Khronos time offset".

Otherwise, resampling is performed. This process spreads Khronos client's queries across servers thereby improving security against strategic and Byzantine attacks (as discussed in Section Section 4.3) and mitigating the effect of a DoS attack on NTP servers that renders them non-responsive. This resampling process continues in subsequent Khronos poll intervals until the two conditions are both satisfied or the number of times the servers are re-sampled exceeds a "Panic Trigger" (K in Table 1), in which case Khronos enters a "Panic Mode". Note that whether the client allows panic mode or not is configurable.

In panic mode, Khronos queries all the servers in its local Khronos pool, orders the collected time samples from lowest to highest and eliminates the lowest third and the highest third of the samples. The client then averages over the remaining samples, and sets this average to be the new "Khronos time offset".

If the Khronos time offset exceeds a predetermined threshold (H) it is passed on to the clock discipline algorithm in order to steer the system time (as in [RFC5905]).

Note that resampling follows immediately the previous sampling since waiting until the next polling interval may increase the time shift in face of attack. This shouldn't generate high overhead since the number of resamples is bounded by K (after K resamplings, "Panic mode" is in place) and the chances to arrive to repeated resampling are low (see Section Section 4 for more details).

3.3. Khronos' Recommended Parameters

According to empirical observations (presented in [Khronos_paper]), querying 15 servers at each poll interval (i.e., m=15) out of 500 servers (i.e., n=500), and setting w to be around 25 ms provides both high time accuracy and good security. Specifically, when selecting w=25ms, approximately 83% of the servers' clocks are at most w-away from UTC, and within 2w from each other, satisfying the first condition of Khronos' system process. However, in order to support congested links scenarios, we recommend to use a higher w value, such as 1 sec.

Furthermore, according to Khronos security analysis, setting K to be 3 (i.e., if after 3 re-samplings the two conditions are not satisfied then Khronos enters "panic mode") is safe when facing time shifting attacks. In addition, the probability of an attacker forcing a panic mode on a client when K equals 3, is negligible (less than 0.000002 for each polling interval).

Khronos' effect on precision and accuracy are discussed in Section 6 and Section 4.

4. Security Considerations

4.1. Threat Model

The following man-in-the-middle (MitM) byzantine attacker is considered: the attacker is assumed to control a subset of the servers in NTP pools and is capable of fully determining the values of the time samples returned by these NTP servers. The threat model encompasses a broad spectrum of MitM attackers, ranging from fairly weak (yet dangerous) MitM attackers only capable of delaying and dropping packets (for example using the Bufferbloat attack) to extremely powerful MitM attackers who are in control of (even authenticated) NTP servers (see detailed security requirements discussion in [RFC7384]).

MitM attackers covered by this model might be, for example, (1) in direct control of a fraction of the NTP servers (e.g., by exploiting a software vulnerability), (2) an ISP (or other Autonomous-System-level attacker) on the default BGP paths from the NTP client to a fraction of the available servers, (3) a nation state with authority over the owners of NTP servers in its jurisdiction, or (4) an attacker capable of hijacking (e.g., through DNS cache poisoning or BGP prefix hijacking) traffic to some of the available NTP servers. The details of the specific attack scenario are abstracted by reasoning about MitM attackers in terms of the fraction of servers with respect to which the attacker has MitM capabilities.

Notably, Khronos provides protection from MitM attacks that cannot be achieved by cryptographic authentication protocols since even with such measures in place an attacker can still influence time by dropping/delaying packets. However, adding an authentication layer (e.g., NTS [RFC8915]) to Khronos will enhance its security guarantees and enable the detection of various spoofing and modification attacks.

4.2. Attack Detection

Khronos detects time-shifting attacks by constantly monitoring NTPv4's (or potentially any other current or future time protocol) clock and the offset computed by Khronos and checking whether the offset exceeds a predetermined threshold (H = 30 ms by default). Unless an attack was detected, NTPv4 controls the client's clock. Under attack, Khronos takes control over the clients clock in order to prevent its shift.

Analytical results (in [Khronos_paper]) indicate that if a local Khronos pool consists of 500 servers, 1/7 of whom are controlled by a man-in-the-middle attacker, and 15 servers are queried in each Khronos poll interval, then success in shifting time of a Khronos client by even a small degree (100 ms), takes many years of effort (over 20 years in expectation). See a brief overview of Khronos' security analysis below.

Khronos' security analysis is briefly described next.

4.3. Security Analysis Overview

Time-samples that are at most w away from UTC are considered "good", whereas other samples are considered "malicious". Two scenarios are considered:

* Less than 2/3 of the queried servers are under the attacker's control.
* The attacker controls more than 2/3 of the queried servers.

The first scenario, where there are more than 1/3 good samples, consists of two sub-cases: (i) there is at least one good sample in the set of samples not eliminated by Khronos (in the middle third of samples), and (ii) there are no good samples in the remaining set of samples. In the first of these two cases (at least one good sample in the set of samples that was not eliminated by Khronos), the other remaining samples, including those provided by the attacker, must be close to a good sample (for otherwise, the first condition of Khronos' system process in Section 3.2 is violated and a new set of servers is chosen). This implies that the average of the remaining samples must be close to UTC. In the second sub-case (where there are no good samples in the set of remaining samples), since more than a third of the initial samples were good, both the (discarded) third lowest-value samples and the (discarded) third highest-value samples must each contain a good sample. Hence, all the remaining samples are bounded from both above and below by good samples, and so is their average value, implying that this value is close to UTC [RFC5905].

In the second scenario, where the attacker controls more than 2/3 of the queried servers, the worst possibility for the client is that all remaining samples are malicious (i.e., more than w away from UTC). However, as proved in [Khronos_paper], the probability of this scenario is extremely low even if the attacker controls a large fraction (e.g., 1/4) of the servers in the local Khronos pool. Therefore, the probability that the attacker repeatedly reach this scenario decreases exponentially, rendering the probability of a significant time shift negligible. We can express the improvement ratio of Khronos over NTPv4 by the ratios of their single shift probabilities. Such ratios are provided in Table Table 2, where higher values indicate higher improvement of Khronos over NTPv4 and are also proportional to the expected time till a time shift attack succeeds once.

Attack Ratio	6 samples	12 samples	18 samples	24 samples	30 samples
1/3	1.93e+01	3.85e+02	7.66e+03	1.52e+05	3.03e+06
1/5	1.25e+01	1.59e+02	2.01e+03	2.54e+04	3.22e+05
1/7	1.13e+01	1.29e+02	1.47e+03	1.67e+04	1.90e+05
1/9	8.54e+00	7.32e+01	6.25e+02	5.32e+03	4.52e+04
1/10	5.83e+00	3.34e+01	1.89e+02	1.07e+03	6.04e+03
1/15	3.21e+00	9.57e+00	2.79e+01	8.05e+01	2.31e+02

Table 2: Khronos Improvement

In addition to evaluating the probability of an attacker successfully shifting time at the client's clock, we also evaluated the probability that the attacker succeeds in launching a DoS attack on the servers by causing many clients to enter a panic mode (and query all the servers in their local Khronos pools). This probability (with the previous parameters of n=500, m=15, w=25 and k=3) is negligible even for an attacker who controls a large number of servers in client's local Khronos pools, and it is expected to take decades to force panic mode.

Further details about Khronos's security guarantees can be found in [Khronos_paper].

5. Khronos' Pseudocode

The pseudocode for Khronos' Time Sampling Scheme, which is invoked in each Khronos poll interval is as follows:

```
counter := 0
S = []
T = []
While counter < K do
   S := sample(m) //gather samples from (tens of) randomly chosen ser
   T := bi_side_trim(S,1/3) //trim the third lowest and highest value
   if (max(T) - min(T) <= 2w) and (|avg(T) - tk| < ERR + 2w) Then
       return avg(T) // Normal case
   end
   counter ++
end
// panic mode
S := sample(n)
T := bi-sided-trim(S,1/3) //trim lowest and highest thirds;
return avg(T)
```

6. Precision vs. Security

Since NTPv4 updates the clock at times when no time-shifting attacks are detected, the precision and accuracy of a Khronos client are the same as NTPv4 at these times. Under attack, Khronos takes control over the client's clock, mitigating the time shift while guaranteeing relatively high accuracy (the error is bounded by H).

Khronos is based on crowdsourcing across servers and regions, changes the set of queried servers more frequently than NTPv4 [Khronos_paper], and avoids some of the filters in NTPv4's system process. These factors can potentially harm its precision. Therefore, a smoothing mechanism can be used, where instead of a simple average of the remaining samples, the smallest (in absolute value) offset is used unless its distance from the average is higher than a predefined value. Preliminary experiments demonstrated promising results with precision similar to NTPv4.

7. Acknowledgements

The authors would like to thank Erik Kline, Miroslav Lichvar, Danny Mayer, Karen O'Donoghue, Dieter Sibold, Yaakov. J. Stein, Harlan Stenn, Hal Murray and Marcus Dansarie, for valuable contributions to this document and helpful discussions and comments.

8. IANA Considerations

This memo includes no request to IANA.

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>.

[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>.

[RFC8915] Franke, D., Sibold, D., Teichel, K., Dansarie, M., and R. Sundblad, "Network Time Security for the Network Time

Protocol", RFC 8915, DOI 10.17487/RFC8915, September 2020, <https://www.rfc-editor.org/info/rfc8915>.

9.2. Informative References

[Khronos_paper] Deutsch, O., Schiff, N.R., Dolev, D., and M. Schapira, "Preventing (Network) Time Travel with Chronos", 2018, <https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_02A-2_Deutsch_paper.pdf>.

Authors' Addresses

Neta Rozen-Schiff
Hebrew University of Jerusalem
Jerusalem
Israel

Phone: +972 2 549 4599
Email: neta.r.schiff@gmail.com

Danny Dolev
Hebrew University of Jerusalem
Jerusalem
Israel

Phone: +972 2 549 4588
Email: danny.dolev@mail.huji.ac.il

Tal Mizrahi
Huawei Network.IO Innovation Lab
Israel

Email: tal.mizrahi.phd@gmail.com

Michael Schapira
Hebrew University of Jerusalem
Jerusalem
Israel

Phone: +972 2 549 4570
Email: schapiram@huji.ac.il