Workgroup: Network Working Group
Internet-Draft: draft-ietf-ntp-chronos-05
Published: 9 July 2022
Intended Status: Informational
Expires: 10 January 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 Chronos

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 Chronos, which is used as a "watchdog" alongside NTPv4, and provides improved security against time shifting attacks. Chronos involves changes to the NTP client's system process only and is backwards compatible with NTPv4 servers. Chronos is also applicable to the emerging NTPv5, since it does not affect the wire protocol.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 10 January 2023.

Copyright Notice

Table of Contents

1. Introduction

NTPv4, as defined in RFC 5905 [RFC5905], is vulnerable to time shifting attacks, in which the attacker's goal is to shift the local time at an NTP client. See [Chronos_paper] for details. 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 determine the response content. This document introduces a time shifting mitigation mechanism called Chronos. Chronos is backwards compatible with NTPv4 and serves as an NTPv4 client's "watchdog" for time shifting attacks. An NTP client that runs Chronos is interoperable with [RFC5905]-compatible NTPv4 servers. Chronos is also applicable to the emerging NTPv5, since it does not affect the wire protocol.

Chronos is a background mechanism that continuously maintains a virtual "Chronos" clock update and compares it to NTPv4's clock update. When the gap between the two updates exceeds a certain threshold (specified in Section 6), this is interpreted as the client experiencing a time shifting attack. In this case, Chronos is used to update the client's clock, and the conventional NTPv4 client algorithm is run in the background until the gap between the two algorithms is again below this threshold, and hence the conventional NTPv4 client algorithm is safe to use again.

Due to Chronos operating in the background, the client clock's precision and accuracy are precisely as in NTPv4 while not experiencing a time-shifting attack. When under attack, Chronos prevents the clock from being shifted by the attacker, thus still preserving high accuracy and precision (as discussed in Section 6).

Chronos achieves accurate synchronization even in the presence of powerful attackers who are in direct control of a large number of NTP servers: up to 1/3 of the servers in the pool (where the pool may consist of hundreds or even thousands of servers). NTPv4 chooses a small subset of the NTP server pool (e.g. 4 servers), and periodically queries this subset of servers. Thus, even if only 1/3 of the servers in the pool are compromised, the small subset that is used by NTPv4 may consist of a majority of faulty servers. Conversely, Chronos constantly updates the set of servers it queries; in each poll interval Chronos randomly chooses a different subset of servers from the pool. Thus, even if an attack is not detected in a given poll interval, Chronos is able to detect the attack within a relatively small number of poll intervals.

A Chronos 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. Chronos is carefully engineered to minimize communication overhead so as to avoid overloading NTP servers. Chronos' security was evaluated both theoretically and experimentally with a prototype implementation. These evaluation results indicate that in order to successfully shift time at a Chronos client by over 100 milliseconds from the UTC, even a powerful man-in-the-middle attacker requires over 20 years of effort in expectation. The full paper is available at [Chronos_paper].

Chronos introduces a watchdog mechanism that is added to the client's system process and maintains a virtual clock value that is used as a reference for detecting attacks. The virtual clock value computation differs from the current NTPv4 in two key aspects. First, a Chronos client relies on a large number of NTP servers, from which only few servers to synchronize with are periodically chosen at random, in order to avoid overloading the servers. Second,

the selection algorithm of the virtual clock uses 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 Chronos' 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].

Selection process Clock filter algorithm and system process [RFC5905].

2.3. Notations

Describing Chronos algorithm, the following notation are used.

Notation	Meaning
n	The number of candidate servers in the pool that Chronos can query (potentially hundreds)
m	The number of servers that Chronos queries in each poll interval (up to tens)
w	An upper bound on the distance of the local time from any NTP server with an accurate clock (termed "truechimer" in [RFC5905])
Cest	The client's estimation for the time that has passed since its last synchronization to the server pool (sec)
B	An upper bound on the client's time estimation error (ms/sec)
ERR	An upper bound on the client's error regarding its estimation of the time passed from the last update, equals to B*Cest (ms)
K	Panic trigger - the number of pool re-sampling until reaches "Panic mode"
tc	The current time [sec], as indicated by the virtual clock value that is computed by Chronos

Table 1: Chronos Notations

The recommended values are discussed in Section 3.2.

3. Extension to the NTP System Process

A client that runs Chronos as a watchdog, uses NTPv4 as in [RFC5905] and in the background runs a modification to the elements of the system process described in Section 11.2.1 and 11.2.2 in [RFC5905] (namely, the Selection Algorithm and the Cluster Algorithm). The NTPv4 conventional protocol periodically queries m servers in each poll interval. In parallel the Chronos watchdog periodically queries a set of m servers in each Chronos poll interval. Specifically, in Chronos, after executing the "Clock Filter Algorithm" as defined in Section 10 in [RFC5905], the client discards outliers by executing the procedure described in this section and the next. Then, the NTPv4 "Combine Algorithm" is used for computing the system peer offset, as specified in Section 11.2.3 in [RFC5905]. In each poll interval the Chronos virtual clock value is compared with the NTPv4 clock value, and if the difference exceeds a predetermined value, an attack is detected. This process holds also for Chronos as a watchdog of future NTPv5.

3.1. Chronos' System Process

At the first time the Chronos system process is executed, calibration is needed. The calibration process generates a local pool of servers the client can synchronize with, consisting of n servers (up to hundreds). To this end, the NTP client executes the "Peer Process" and "Clock Filter Algorithm" as in Sections 9,10 in [RFC5905] (respectively), on an hourly basis, for 24 consecutive hours, and generates the union of all received NTP servers' IP addresses. Importantly, this process can also be executed in the background periodically, once in a long time (e.g., every few weeks/ months).

In each Chronos poll interval the Chronos system process randomly chooses a set of m servers (where n with magnitude of hundreds and m of tens) out of the local pool of n servers. Then, out of the time-samples received from this chosen subset of servers, a lowest third of the samples' offset values and highest third of the samples' offset values are discarded.

Chronos checks that the following two conditions hold for the remaining samples:

- The maximal distance between every two time samples does not exceed 2w.

- The average value of the remaining samples is at distance at most ERR+2w from the client's local clock (as computed by Chronos).

(where w, ERR are as described in Table 1. Notice that ERR magnitude is approximately LAMBDA as defined in [RFC5905]).

In the event that both of these conditions are satisfied, the average of the remaining samples is the "final offset". Otherwise, a random partial of the interval is chosen, after which a new subset of servers is sampled, in the exact same manner. This way, Chronos client queries are spread across the time interval better in case of DoS attack on the NTP servers. This resampling process continues in subsequent Chronos 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, Chronos enters a "Panic Mode". Note that it is configurable whether the client allows panic mode or not.

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

As in [RFC5905], the final offset is passed on to the clock discipline algorithm for the purpose of steering the Chronos virtual clock to the correct time. The Chronos virtual clock is then compared to the NTPv4 (or to the future NTPv5) clock as part of the watchdog process.

3.2. Chronos' Recommended Parameters

According to empirical observations (presented in [Chronos_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 milliseconds provides both high time accuracy and good security. Moreover, empirical analyses showed that,on average, when selecting w=25ms, approximately 83% of the servers' clocks are at most w-away from the UTC, and within 2w from each other, satisfying the first condition of Chronos' system process.

Furthermore, according to Chronos security analysis, setting K to be 3 (i.e., if after 3 re-sampling, the two conditions are not satisfied, then Chronos reaches "panic mode") is both safe when facing time shifting attacks and the probability of reaching the "panic mode" is negligible (less than 0.000002).

Chronos effect on precision and accuracy are discussed in Section 5 and Section 6.

4. Chronos' Pseudocode

The pseudocode for Chronos' Time Sampling Scheme, which is invoked in each Chronos 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)-tc| < ERR + 2w) Then
       return avg(t)
   end
   counter ++
   sleep(rand(0,1)*poll interval)
end
// panic mode
S := sample(n)
T := bi-sided-trim(S,1/3) //trim bottom and top thirds;
return avg(T)
```

5. Precision vs. Security

Since NTPv4 (and future NTPv5) updates the clock as long as time-shifting attacks are not detected, the precision and accuracy of a Chronos client are the same as NTPv4 when not under attack. Under attack, Chronos, changes the list of the sampled servers more frequently than NTPv4 [Chronos_paper], and does not use some of the filters in NTPv4's system process, can potentially be less precise (though provably more secure than NTPv4, which is vulnerable to time-shifting attacks [RFC5905]).

6. Security Considerations

As explained above, Chronos repeatedly gathers time samples from small subsets of a large local pool of NTP servers. The following form of a man-in-the-middle (MitM) Byzantine attacker is considered: the MitM attacker is assumed to control a subset of the servers in the local pool of servers and is capable of determining precisely the values of the time samples gathered by the Chronos client from these NTP servers. The threat model thus encompasses a broad spectrum of MitM attackers, ranging from fairly weak (yet dangerous) MitM attackers only capable of delaying and dropping packets to extremely powerful MitM attackers who are in control of (even authenticated) NTP servers. MitM attackers captured by this framework 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.

Chronos detects time-shifting attacks by constantly monitoring NTPv4's (or NTPv5's) offset and the offset computed by Chronos, as explained above, and checking whether it exceeds a certain threshold (10 milliseconds by default).

Analytical results (in [Chronos_paper]) indicate that in order to succeed in shifting time at a Chronos client by even a small amount (e.g., 100 milliseconds), even a powerful MitM attacker requires many years of effort (e.g., over 20 years in expectation). See a brief overview of Chronos' security analysis below.

Notably, Chronos 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 and crypto-based security layer to Chronos will enhance its security guarantees and enable the detection of various spoofing and modification attacks.

Chronos' security analysis is briefly described next.

6.1. Security Analysis Overview

Time-samples that are at most w away from the 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 Chronos (for example, 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 Chronos), the other remaining samples, including those provided by the attacker, must be close to a good sample (for otherwise, the

first condition of Chronos' system process in Section 3.1 is violated and a new set of servers is chosen). This implies that the average of the remaining samples must be close to the 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 the 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 the UTC). However, as proved in [Chronos_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 pool. Therefore, the probability that the attacker repeatedly succeeds in realising this scenario decreases exponentially, rendering the probability of a significant time shift negligible. See [Chronos_paper] for details.

Beyond 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 panic mode (and so query all the servers in their local pools). This probability (with the previous parameters of n=500, m=15, w=25 and k=30) is negligible even for an attacker in control of a large number of servers in clients' local server pools, and it will take attacker decades to force panic mode.

Further details about Chronos's security considerations can be found in [Chronos_paper].

7. Acknowledgements

The authors would like to thank Erik Kline, Miroslav Lichvar, Danny Mayer, Karen O'Donoghue, Dieter Sibold, Yaakov. J. Stein, and Harlan Stenn, 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>.

9.2. Informative References

[Chronos_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