Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 30, 2018

N. R.Schiff
D. Dolev
Hebrew University of Jerusalem
T. Mizrahi
Marvell
M. Schapira
Hebrew University of Jerusalem

June 28, 2018

A Secure Selection and Filtering Mechanism for the Network Time Protocol Version 4

draft-schiff-ntp-chronos-00

Abstract

The Network Time Protocol version 4 (NTPv4) defines the peer process, the clock filter algorithm, the system process and the clock description algorithm. The clock filter algorithm and the system process, as defined in RFC 5905, are the mechanism according to which an NTP client chooses the NTP servers it synchronized with. This document specifies an alternative set of client mechanisms, named Chronos, that is backward compatible with NTPv4, and offers an improved level of security against time shifting attacks.

Status of This Memo

This Internet-Draft will expire on December 30, 2018.

Copyright Notice

Table of Contents

1. Introduction

According to RFC 5905 [RFC5905], the NTP servers used for updating the client's time are chosen by the clock filter algorithm and the system process. However, this method may be vulnerable to time shifting attacks, in which the attacker's goal is to shift the local time of an NTP client. Time shifting attacks on NTP are possible even if all NTP communications are encrypted and authenticated. This document introduces an improved system process with a secure algorithm called Chronos. Chronos is backwards compatible with NTPv4, as an NTP client that runs Chronos is interoperable with [RFC5905]-compatible NTPv4 servers.

Chronos achieves accurate synchronization even in the presence of powerful attackers who are in direct control of a large number of NTP servers. Chronos leverages ideas from distributed computing literature on clock synchronization in the presence of adversarial (Byzantine) behaviour.

A Chronos client iteratively "crowdsources" time queries across multiple NTP servers and applies a provably secure algorithm for eliminating "suspicious" responses and 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. The experimental results indicate that in order to implement a successful time-shifting attack on a Chronos client by over 100ms from the UTC, even a powerful man-in-the-middle attacker requires over 20 years of effort in expectation. The full paper is in [Chronos_paper].

Chronos differs from the current NTPv4 in two aspects. First, the Chronos client relies on a large number of NTP servers, from which only few are chosen at random in order to avoid overloading the servers. Second, the selection algorithm uses an approximate agreement technique to remove outliers, thus limiting the attacker's ability to contaminate the chosen time samples. These Chronos client mechanisms have provable security guarantees against man-in-the-middle attackers and attackers who are 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 RFC 2119 [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.

Notaion	Meaning
w	An upper bound on the distance from the local time at any NTP server with an accurate clock ("truechimer" as in [RFC5905])
Cest	the client's estimate for the time that passed since its last synchronization to the server pool (sec)
ERR	(2W*Cest)/1000
K	panic trigger
tc	the current time, as indicated by the client's local clock [sec]

Table 1: Chronos Notations

3. Extension for NTP Selection Process

A client that runs Chronos does not implement the functionality described in Sections 10 and 11 in [RFC5905]. Instead, the client implements the behavior described in this section and the next one.

3.1. Peer calibration Process

The peer calibration process gathers a server pool of hundreds of servers. Each NTP client conducts the peer process as in Section 9 in [RFC5905], on an hourly basis for 24 consecutive hours and generates the union of all received IP addresses. Importantly, this is executed in the background once in a long time (e.g., every few weeks/months).

3.2. Chronos Selection Process

The Chronos selection process samples the server pool and removes outliers (replaces the clock filter algorithm and the system process as in [RFC5905]). First, a subset on the order of tens of the servers in the server pool is selected at random. Then, out of the tens of collected samples, the third lowest-value samples and third highest value samples are discarded.

Given the remaining samples, Chronos checks two conditions:

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

- The average value of the remaining samples is at a distance of at most ERR+2w from the client's local clock.

(where w,ERR are described in Table 1).

In the event that both of these conditions are satisfied, the average of the remaining samples is the "final offset". Otherwise, a few tens of the servers from the pool are sampled again, in the exact same manner. This re-sampling process continues until the two conditions are finally 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".

In panic mode a Chronos client queries all the servers in the 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, which become the new "final offset".

As in [RFC5905], the final offset is passed to the clock discipline algorithm to steer the system clock to the correct time.

4. Chronos Pseudocode

The Chronos pseudocode Time Sampling Scheme is the following:

```
counter := 0
While counter < K do
    S := sample(m) //gather sample from tens randomly chosen servers
    T := bi-side-trim(S,1/3) //trim third lowest and highest values
    if (max(T) -min(T) <= 2w) and (|avg(T)-tc| < ERR + 2w) Then
        return avg(t)
    end
counter ++;
end
// panic mode;
S := sample(n);
T := bi-sided-trim(S,n/3) //trim bottom and top thrids;
return avg(T)
```

5. Acknowledgements

...

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

As explained above, a Chronos client repeatedly gathers time samples from small subsets of a large pool of NTP servers. The following form of a man-in-the-middle (MitM) Byzantine attacker is considered: a MitM attacker is assumed to control a subset of the servers in the pool of available 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 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.

Analytical results (in [Chronos_paper]) indicate that in order to succeed in shifting time at a Chronos client by even a small time shift (e.g., 100ms), even a powerful man-in-the-middle attacker requires many years of effort (e.g., over 20 years in expectation).

It should be noted that Chronos provides resilience to MitM attacks that cannot be achieved by cryptographic authentication protocols. However, adding an authentication and crypto-based security layer to the Chronos layer is important for achieving high security guarantees and detection of various spoofing and modification attacks.

Further details about the Chronos security considerations and guarantees are discussed in [Chronos_paper].

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

8.2. Informative References

[Chronos_paper] Deutsch, O., Schiff, N., Dolev, D., and M. Schapira, "Preventing (Network) Time Travel with Chronos", 2018, <http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_02A-2_Deutsch_paper.pdf>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <https://www.rfc-editor.org/info/rfc2629>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

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
Marvell
6 Hamada St.
Yokneam 2066721
Israel

Email: talmi@marvell.com

Michael Schapira
Hebrew University of Jerusalem
Jerusalem
Israel

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