Workgroup: t2trg
Internet-Draft: draft-amsuess-t2trg-raytime-01
Published: 8 July 2023
Intended Status: Experimental
Expires: 9 January 2024
Authors: C. Amsüss

Raytime: Validating token expiry on an unbounded local time interval

Abstract

When devices are deployed in locations with no real-time access to the Internet, obtaining a trusted time for validation of time limited tokens and certificates is sometimes not possible. This document explores the options for deployments in which the trade-off between availability and security needs to be made in favor of availability. While considerations are general, terminology and examples primarily focus on the ACE framework.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-amsuess-t2trg-raytime/.

Discussion of this document takes place on the Thing-to-Thing Research Group mailing list (mailto:t2trg@irtf.org), which is archived at https://mailarchive.ietf.org/arch/browse/t2trg/. Subscribe at https://www.ietf.org/mailman/listinfo/t2trg/.

Source for this draft and an issue tracker can be found at https://gitlab.com/chrysn/raytime.

Status of This Memo

Table of Contents

1. Introduction

[See also abstract.]

1.1. Terminology

This document uses terminology from the ACE framework ([ACE]) to describe the typical participants, even in general parts that may apply beyond ACE. In particular:

* The Resource Server (RS) is a constrained device, also sometimes called “the device”.

  In the context of this document, it has no regular access to the Internet, and its ability to maintain a concept of local time is limited by intermittent power supplies.

* The Authorization Server (AS) is a party trusted by the RS.

  In the context of this document, it is connected to the Internet. (When more private networks are used, the term Internet can be

replaced with the partition of the private networks that contains the AS). It is considered the source of trusted time in this document, as distributed time sources are not relevant to its mechanisms.

* The Client is a device that has obtained time- and scope limited credentials for the RS from the AS.

  In the context of this document, it acquires these credentials ahead of time before it moves out of communication range of the Internet and into communication range of the RS.

  Beyond these credentials, the client is untrusted in the sense that no action of the RS should allow it to exceed the permissions encoded in the credentials it holds.

1.2. Motivating example

Devices that use Internet style connectivity and security have been deployed far beyond the range of easily accessible connectivity, for example in remote forests [forest40] and in space [space].

For a concrete example, consider a scientific instrument set up by students. The instrument is under the administrative control of the university, which is issuing time limited access tokens that allow configuring experiments and reading data, but not altering inventory designations or handing off ownership.

The instrument is set up in a far-off valley without cellular reception, and visited by students once per week to collect data, verify that it is still operational, and perform adjustments or repairs on site as needed. Any repairs require powering down the device completely.

When following the mechanisms and guidance of this document, the instrument will be usable by the students even after they had to power it down completely, and (once used by the new students) justifiedly reject the tokens of students that used the device earlier. It will stay usable to malicious students if they disassemble it and use it after their allowed time on the instrument has expired, but can only be operated for a total time of about the validity time span of those users' tokens.

1.3. Security and availability

When a device has no current time, no means of acquiring time, and receives a time limited token to authorize an action, two outcomes are possible: the action is rejected or allowed. Rejection is the safe alternative in terms of security, but reduces the availability of the device, possibly to the point of not providing its function

at all. Conversely, allowing the action maintains availability at the risk of violating security objectives.

The fundamental trade-off between security and availability is common around time limited access, and usually manifested in a choice of expiry times: A token with a validity of one week (or, equivalently, a revocation list with a maximum age of one week) will be usable across a day-long outage of the authorization infrastructure. On the downside, it provides technical authorization to an administratively deauthorized user for several days after revocation of permissions.

This document concerns itself with getting the most security out of the case that favors availability after a period of no operational clock.

1.4. Threat model

Three main threats are considered here:

* Using old tokens.

  A user who has obtained a token may attempt to use it after the token's lifetime (and the user's permission) has expired.

* Manipulation of time sources.

  An attacker controlling a time source which the device trusts may make false statements on the current time. Indicating an earlier time can be used to extend the usability of old tokens. Indicating a later time can be used to deny service to users of valid tokens.

* Manipulation of the clock.

  An attacker with physical access (possibly only to the device's environment) may cut power to the device's clock.

Other attacks, such as physical attacks on the device, or exploiting weaknesses of the cryptographic systems used, are relevant to the system, but have no different impact on a system following this document than on a system that requires an operational clock before it allows access.

1.5. Applicability constraints

The mechanisms of this document are intended only for scenarios matching the listed prerequisites. If either of them are not met,

there are better solutions available, with suggestions listed along the condition.

* There is no network connectivity, not even for the client.

  If there is, the RS may use communication with the AS to obtain a current time value (as described in [ace-time]), perform token validation ([ACE]). Alternatively, it may use other Internet services such as [roughtime].

  Note that due to CoAP's good proxy support, the connectivity between the RS and the AS does not need to be in a continuous IP network.

* The lifetime of tokens is limited.

  When tokens of unlimited lifetime are used, there is no need to employ the methods described in this document.

* Devices have no means of reliably maintaining time throughout their whole lifetime.

  If the power supply of the device's clock is reliable enough to be sure to outlive the device, regular evaluation of time bounds can be done, possibly accounting for an upper bound of clock skew.

* Devices need to stay accessible after local power interruptions, that is, favor availability over security after a loss of local time.

  Otherwise, the device can issue an AS Request Creation Hint containing a cnonce (described in [ACE]). The client then needs to travel back to into communication distance to the AS, and obtain a token confirming that cnonce. Alternatively, one of the time synchronization mechanisms mentioned above can be used.

The applicability is also limited when it comes to requirements of a certain date being in the past, such as "nbf" ("not before") claims described in [JWT]. These claims are notoriously hard to verify after a loss of time (as discussed in [imperfect]), but fortunately also rarely needed. The use of such claims is not fundamentally prohibitive of this document's raytime mechanism, but its approach favoring availability over security is inapplicable to them (as a device would accept any nbf value after power-up).

While data sources such as radio time signal stations (e.g., DCF77, MSF, WWVB) or GNSSs such as GPS are frequently offered as a solution, neither is reliable enough for use with security systems:

Radio time signals are effectively unauthenticated, and GPS signals can be forged with hardware that is now cheap [gps].

2. Raytime

It is relatively common in clock synchronization to treat known time in interval arithmetic as the earliest and latest possible current point in time (for an example, see [optimal]).

When a device has been dormant for an indeterminate amount of time, that interval's upper bound needs to be set to infinity, creating a half unbounded interval, or a ray. For lack of a term established in literature [i.e., if you happen to find one, please tell and this will change], we call devices operating under such conditions to work "in raytime". A device in raytime can be sure that some points in time have passed, but never that they have not.

Devices that require a two-sided bound on some credentials they accept may use any trusted time synchronization mechanism to establish an upper bound on current time and switch to interval time (and fall back to raytime after the next loss of continuous time). Devices that never evaluate the upper bound on time anyway may only implement the lower bound, and always operate in raytime.

Maintaining raytime is fundamentally a best-effort undertaking. Implementations have two mechanisms to advance time:

* Any time the device receives a trusted statement on a time stamp being in the past that is ahead of its earliest possible current time, it updates that bound.

  Care has to be taken to limit this mechanism to trusted time sources. It is recommended to exclusively use statements from the AS, as they are provided in tokens' "iat" claim. The build time of the latest firmware update may also provide such a time source, provided it is known to be on the same timescale as the AS's timescale.

* As time passes, the device advances the lower bound on the current time. To avoid refusing tokens used quickly after issuance, lower bound time should be advanced slower than time actually passes, accounting for the maximum specified clock skew.

Both kinds of advancements need to be recorded persistently, lest power cycling the device makes all tokens valid under raytime assumptions. To avoid wearing out limited persistence options, writing may be delayed as suitable in the application. The trade-off to consider here are flash write cycles and power consumption of flash writes versus the additional time malicious users gain for using the device by removing the power supply at the right time.

Being best-effort, there is no need to transactionally commit the time seen in tokens – this is not replay protection.

3. Using raytime with ACE

[This section will contain concrete guidance when the open questions are resolved.]

3.1. Open questions

* Should the decision on whether raytime is OK to use be encoded in the token or in the application?

  Options are to use “exp” and say that the action will tell whether or not raytime is good enough to evaluate the credentials’ validity (some actions may require more assurances), or to use an “exp-besteffort” or similar indicator to describe the whole token. The latter approach may be limiting in that then not all permissions can be expressed in a single token, but then again, that’s also true with different validity times.

  More generally (also for interval time), can a token indicate whether the clock skew upper bound is used in favor or against the user? [JWT] talks of “some small leeway”, but is not overly precise.

* If the device *is* used online, can it set a cnonce in its creation hints and the application will still try to use a cnonce-less token? Or should optional cnonces use a different key? (Setting a cnonce in online situations does have the advantage that the device can convert from raytime to interval time. But is that even useful, unless the tokens used have nbf claims?)

4. Security Considerations

5. Informative References

[ACE] Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments Using the OAuth 2.0 Framework (ACE-OAuth)", RFC 9200, DOI 10.17487/RFC9200, August 2022, <https://www.rfc-editor.org/rfc/rfc9200>.

[ace-time] Navas, R., Selander, G., and L. Seitz, "Lightweight Authenticated Time (LATe) Synchronization Protocol", Work in Progress, Internet-Draft, draft-navas-ace-secure-time-synchronization-00, 31 October 2016, <https://datatracker.ietf.org/doc/html/draft-navas-ace-secure-time-synchronization-00>.

[forest40] Singh, R., Gehlot, A., Vaseem Akram, S., Kumar Thakur, A., Buddhi, D., and P. Kumar Das, "Forest 4.0: Digitalization of forest using the Internet of Things (IoT)", Journal of King Saud University - Computer and Information Sciences vol. 34, no. 8, pp. 5587-5601, DOI 10.1016/j.jksuci.2021.02.009, September 2022, <https://doi.org/10.1016/j.jksuci.2021.02.009>.

[gps] Sharp, J., Wu, C., and Q. Zeng, "Authentication for drone delivery through a novel way of using face biometrics", Proceedings of the 28th Annual International Conference on Mobile Computing And Networking, DOI 10.1145/3495243.3560550, October 2022, <https://doi.org/10.1145/3495243.3560550>.

[imperfect] Carrol, J. L. and D. B. Carren, "Future Imperfect", Star Trek: The Next Generation , 1990.

[JWT] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/rfc/rfc7519>.

[optimal] Schmid, U. and K. Schossmaier, "Interval-based clock synchronization with optimal precision", Information and Computation vol. 186, no. 1, pp. 36-77, DOI 10.1016/s0890-5401(03)00103-2, October 2003, <https://doi.org/10.1016/s0890-5401(03)00103-2>.

[roughtime] Malhotra, A., Langley, A., Ladd, W., and M. Dansarie, "Roughtime", Work in Progress, Internet-Draft, draft-ietf-ntp-roughtime-07, 26 September 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-ntp-roughtime-07>.

[space] Kua, J., Arora, C., Loke, S., Fernando, N., and C. Ranaweera, "Internet of Things in Space: A Review of Opportunities and Challenges from Satellite-Aided Computing to Digitally-Enhanced Space Living", arXiv article, DOI 10.48550/ARXIV.2109.05971, 2021, <https://doi.org/10.48550/ARXIV.2109.05971>.

Appendix A. Change log

Since -00:

* Write 'raytime' as a single word rather than 'ray time'.

* Editorial fixes provided by Carsten Bormann.

Acknowledgments

TBD: CB

Author's Address

Christian Amsüss
Austria

Email: christian@amsuess.com