ACE Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2020

M. Tiloca
RISE AB
L. Seitz
Combitech
F. Palombini
Ericsson AB
S. Echeverria
G. Lewis
CMU SEI

March 09, 2020

Notification of Revoked Access Tokens in the Authentication and Authorization for Constrained Environments (ACE) Framework draft-tiloca-ace-revoked-token-notification-01

Abstract

This document specifies a method of the Authentication and Authorization for Constrained Environments (ACE) framework, which allows an Authorization Server to notify Clients and Resource Servers (i.e., registered devices) about revoked Access Tokens. The method relies on resource observation for the Constrained Application Protocol (CoAP), with Clients and Resource Servers observing a Token Revocation List on the Authorization Server. Resulting unsolicited notifications of revoked Access Tokens complement alternative approaches such as token introspection, while not requiring additional endpoints on Clients and Resource Servers.

Table of Contents

1. Introduction

Authentication and Authorization for Constrained Environments (ACE) [I-D.ietf-ace-oauth-authz] is a framework that enforces access control on IoT devices acting as Resource Servers. In order to use ACE, both Clients and Resource Servers have to register with an Authorization Server and become a registered device. Once registered, a Client can send a request to the Authorization Server for an Access Token for a Resource Server. For a Client to access the Resource Server, the Client must present the issued Access Token at the Resource Server, which then validates and stores it.

Even though Access Tokens have expiration times, there are circumstances by which an Access Token may need to be revoked before its expiration time, such as: (1) a registered device has been compromised, or is suspected of being compromised; (2) a registered device is decommissioned; (3) there has been a change in access policies for a registered device; and (4) there has been a change in the ACE profile for a registered device.

As discussed in Section 6.1 of [I-D.ietf-ace-oauth-authz], only client-initiated revocation is currently specified [RFC7009] for OAuth 2.0, based on the assumption that Access Tokens in OAuth are issued with a relatively short lifetime. However, this may not be the case for constrained, intermittently connected devices, that need Access Tokens with relatively long lifetimes.

This document specifies a method for allowing registered devices to access and observe a Token Revocation List (TRL) resource on the Authorization Server, in order to get an updated list of revoked, but yet not expired, pertaining Access Tokens. In particular, registered devices rely on resource observation for the Constrained Application Protocol (CoAP) [RFC7641]. The benefits of this method are that it complements token introspection and does not require any additional endpoints on the registered devices.

1.1. Terminology

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.

Readers are expected to be familiar with the terms and concepts described in the ACE framework for Authentication and Authorization [I-D.ietf-ace-oauth-authz], as well as with terms and concepts related to CBOR Web Tokens (CWTs) [RFC8392], and JSON Web Tokens (JWTs) [RFC7519]. The terminology for entities in the considered architecture is defined in OAuth 2.0 [RFC6749]. In particular, this includes Client, Resource Server, and Authorization Server.

Readers are also expected to be familiar with the terms and concepts related to CBOR [RFC7049], JSON [RFC8259], the CoAP protocol [RFC7252], CoAP Observe [RFC7641], and the use of hash functions to name objects as defined in [RFC6920].

Note that, unless otherwise indicated, the term "endpoint" is used here following its OAuth definition, aimed at denoting resources such as /token and /introspect at the Authorization Server, and /authz-

info at the Resource Server. This document does not use the CoAP definition of "endpoint", which is "An entity participating in the CoAP protocol."

This specification also refers to the following terminology.

- Token hash: identifier of an Access Token, in binary format encoding. The token hash has no relation to other possibly used token identifiers, such as the "cti" (CWT ID) claim of CBOR Web Tokens (CWTs) [RFC8392].

- Token Revocation List (TRL): a collection of token hashes, in which the corresponding Access Tokens have been revoked but are not expired yet.

- TRL resource: a resource on the Authorization Server, with a TRL as its representation.

- TRL endpoint: an endpoint at the Authorization Server associated to the TRL resource. The default name of the TRL endpoint in a url-path is '/revoke/trl'. Implementations are not required to use this name, and can define their own instead.

- Registered device: a device registered at the Authorization Server, as a Client, a Resource Server, or both. A registered device acts as caller of the TRL endpoint.

- Administrator: entity authorized to get full access to the TRL at the Authorization Server, and acting as caller of the TRL endpoint. An administrator is not necessarily a registered device as defined above, i.e. a Client requesting Access Tokens or a Resource Server consuming Access Tokens. How the administrator authorization is established and verified is out of the scope of this specification.

- Pertaining Access Token:

  * With reference to an administrator, an Access Token issued by the Authorization Server.

  * With reference to a registered device, an Access Token intended to be owned by that device. An Access Token pertains to a Client if the Authorization Server has issued the Access Token and provided it to that Client. An Access Token pertains to a Resource Server if the Authorization Server has issued the Access Token to be consumed by that Resource Server.

2. Protocol Overview

This protocol defines how a CoAP-based Authorization Server informs Clients and Resource Servers, i.e. registered devices, about revoked Access Tokens. How the relationship between the registered device and the Authorization Server is established is out of the scope of this specification.

At a high level, the steps of this protocol are as follows.

- Upon startup, the Authorization Server creates a TRL resource. At any point in time, the TRL resource represents the list of all revoked Access Tokens issued by the Authorization Server that are yet not expired.

- When a device is registered at the Authorization Server, it receives the url-path to the TRL resource. After the registration procedure is finished, the registered device sends an Observation Request to that TRL resource as described in [RFC7641], i.e. a GET request with an Observe option set to 0 (register). Upon receiving the request, the Authorization Server adds the registered device to the list of observers of the TRL resource. At any time, the registered device can send a GET request to the TRL endpoint, in order to get the current list of pertaining revoked Access Tokens.

- When an Access Token is revoked, the Authorization Server adds the corresponding token hash to the TRL. Also, when a revoked Access Token eventually expires, the Authorization Server removes the corresponding token hash from the TRL. In either case, after updating the TRL, the Authorization Server sends Observe Notifications as described in [RFC7641]. That is, one Observe Notification is sent to each registered device the Access Token pertains to, and specifies the current updated list of token hashes in the portion of the TRL pertaining to that device.

- An administrator can observe and access the TRL like a registered device, while getting the full updated representation of the TRL.

Figure 1 provides a high-level overview of the service provided by this protocol. Each dotted line associated to a pair of registered devices indicates the Access Token that they both own. In particular, Figure 1 shows the Observe Notifications sent by the Authorization Server to four registered devices and one administrator, upon revocation of the issued Access Tokens t1, t2 and t3, with token hash th1, th2 and th3, respectively.

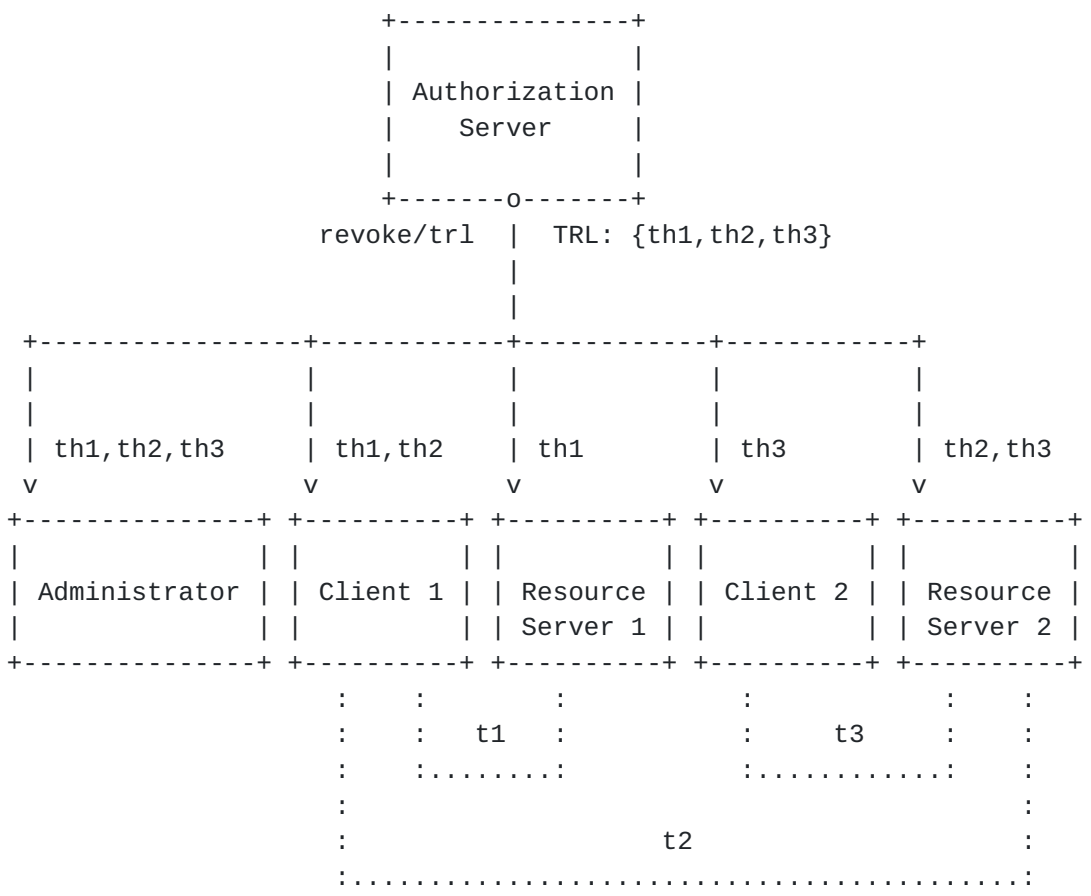

Figure 1: Protocol Overview

A more detailed example describing the protocol flow and message exchange between the Authorization Server and a registered device is provided in Section 8.

3. Token Hash

The token hash of an Access Token is generated as follows.

1. The Authorization Server defines ENCODED_TOKEN, as the value of the 'access_token' field from the Authorization Server response where that Access Token was included and returned from the /token endpoint to the requesting Client.

2. The Authorization Server defines HASH_INPUT as:

   * If CBOR was used to transport the Access Token, the byte-string value of ENCODED_TOKEN.

   * If JSON was used to transport the Access Token, the binary representation of the String value of ENCODED_TOKEN, which would depend on the used charset.

3. The Authorization Server generates a hash value of HASH_INPUT as per Section 6 of [RFC6920]. The resulting output in binary format is used as the token hash.

The specifically used hash-function MUST be collision-resistant on byte-strings, and MUST be selected from the "Named Information Hash Algorithm" Registry defined in Section 9.4 of [RFC6920].

4. The TRL Resource

Upon startup, the Authorization Server creates a single TRL resource.

The initial content of the TRL resource representation MUST be an empty CBOR array, i.e. the TRL is initialized as empty.

The order of the token hashes in the CBOR array is irrelevant, and the CBOR array MUST be treated as a set in which the order has no significant meaning.

4.1. Update of the TRL Resource

The Authorization Server updates the TRL in the following two cases.

- When a non-expired Access Token is revoked, the token hash of the Access Token is added to the TRL resource representation. That is, the token hash is added to the CBOR array used as TRL resource representation.

- When a revoked Access Token expires, the token hash of the Access Token is removed from the TRL resource representation. That is, the token hash is removed from the CBOR array used as TRL resource representation.

5. The TRL Endpoint

Consistently with Section 6.5 of [I-D.ietf-ace-oauth-authz], all communications between a caller of the TRL endpoint and the Authorization Server MUST be encrypted, integrity and replay protected. Furthermore, responses from the Authorization Server to the caller MUST be bound to the caller's request.

The Authorization Server MUST implement measures to prevent access to the TRL endpoint by entities other than registered devices and authorized administrators.

The TRL endpoint supports only the GET method, and provides two types of query of the TRL.

- Full query: the Authorization Server returns the token hashes of the revoked Access Tokens currently in the TRL and pertaining to the issuer of the GET request. The processing of a full query and the related response format are defined in Section 5.1.

- Diff query: the Authorization Server returns a set of diff entries. Each entry is related to one of the most recent updates, in the portion of the TRL pertaining to the issuer of the GET request. In particular, the entry associated to one of such updates contains a list of token hashes, such that i) the corresponding revoked Access Tokens pertain to the issuer of the GET request; and ii) they were added to or removed from the TRL at that update. The processing of a diff query and the related response format are defined in Section 5.2.

The TRL endpoint admits the following query parameters in a GET request.

- 'diff': if included, its value MUST be set to "true" and indicates to perform a diff query of the TRL.

- 'N': if included, its value MUST be a positive integer greater than 0. This parameter is relevant when requesting to perform a diff query of the TRL, and indicates the maximum number of diff entries that a (notification) response should include. This parameter MUST NOT be present if the 'diff' query parameter is not present.

5.1. Full Query of the TRL

In order to produce a (notification) response to a GET request asking for a full query of the TRL, the Authorization Server performs the following actions.

1. The Authorization Server builds from the current TRL resource representation a set HASHES of token hashes, such that:

   * If the issuer of the GET request is a registered device, HASHES includes the token hashes of the Access Tokens pertaining to that registered device. The Authorization Server can use the authenticated identity of the registered device to perform the necessary filtering on the TRL resource representation.

   * If the issuer of the GET request is an administrator, HASHES includes all the token hashes in the current TRL resource representation.

2. The Authorization Server prepares a 2.05 (Content) Response for the GET request issuer, with a CBOR Array as payload. Each element of the array specifies one of the token hashes from the set HASHES.

   The order of the token hashes in the CBOR array is irrelevant, i.e. the CBOR array MUST be treated as a set in which the order has no significant meaning.

5.2. Diff Query of the TRL

In order to produce a (notification) response to a GET request asking for a diff query of the TRL, the Authorization Server performs the following actions.

1. The Authorization Server defines the positive integer SIZE. If the GET request did not include the query parameter N, or N was greater than a pre-defined positive integer N_MAX, SIZE takes the value of N_MAX. Otherwise, SIZE takes the value of N.

2. The Authorization Server prepares U <= SIZE diff entries. The entries are related to the latest U updates that affect the portion of the TRL pertaining to the issuer of the GET request. In particular, the first entry refers to the most recent of such updates, the second entry refers to the second from last of such updates, and so on. Each diff entry is a CBOR Map, which includes the following two elements.

   * The first element has label "removed" and a CBOR Array as value. Each element of the array is the token hash of an Access Token, that pertained to the issuer of the GET request and that was removed from the TRL during the update associated to the diff entry.

   * The second element has label "added" and a CBOR Array as value. Each element of the array is the token hash of an Access Token, that pertains to the issuer of the GET request and that was added to the TRL during the update associated to the diff entry.

3. The Authorization Server prepares a 2.05 (Content) Response for the issuer of the GET request, with a CBOR Array of U elements as payload. Each element of the array specifies one of the CBOR Maps prepared at point 2 as diff entries.

Within the CBOR Array, the CBOR Maps are sorted to reflect the corresponding updates to the TRL in reverse chronological order. That is, the first CBOR Map relates to the most recent update to the portion of the TRL pertaining to the issuer of the GET request.

However, in each of the CBOR Maps, the order of the token hashes in the CBOR arrays "removed" and "added" is irrelevant, i.e. those CBOR arrays MUST be treated as a set in which the order has no significant meaning.

If the Authorization Server supports diff queries:

- The Authorization Server MUST keep track of N_MAX latest updates to the portion of the TRL that pertains each caller of the TRL endpoint. The particular method to achieve this is implementation-specific.

- The Authorization Server SHOULD provide registered devices and administrators with the value of N_MAX, upon their registration (see Section 6).

If the Authorization Server does not support diff queries, it proceeds as when processing a full query (see Section 5.1).

6. Upon Registration

During the registration process at the Authorization Server, an administrator or a registered device receives the following information as part of the registration response.

- The url-path to the TRL endpoint at the Authorization Server.

- Optionally, a positive integer N_MAX, if the Authorization Server supports diff queries of the TRL resource (see Section 5.2).

After the registration procedure is finished, the administrator or registered device performs a GET request to the TRL resource, including the CoAP Observe option set to 0 (register), in order to start an observation of the TRL resource at the Authorization Server, as per Section 3.1 of [RFC7641]. The GET request can express the wish for a full query (see Section 5.1) or a diff query (see Section 5.2).

The Authorization Server MUST reply using the CoAP response code 2.05 (Content) and the CoAP Observe option with value 1. The response payload is formatted as defined in Section 5.1 or in Section 5.2, in

case the GET request was for a full query or a diff query of the TRL, respectively.

7. Notification of Revoked Tokens

In the case the TRL is updated (see Section 4.1), the Authorization Server sends Observe Notifications to every observer of the TRL resource. Observe Notifications are sent as per Section 4.2 of [RFC7641].

The content of each Observe Notification is formatted as defined in Section 5.1 or in Section 5.2, in case the original Observation Request was for a full query or a diff query of the TRL, respectively.

Furthermore, an administrator or a registered device can send additional GET requests to the TRL endpoint at any time, in order to retrieve the token hashes of the pertaining revoked Access Tokens. When doing so, the caller of the TRL endpoint can perform a full query (see Section 5.1) or a diff query (see Section 5.2).

8. Example

Figure 2 shows an example interaction between a Resource Server RS and an Authorization Server AS, considering a CoAP observation and a full query of the TRL.

The details of the registration process are omitted, but it is assumed that the Resource Server sends an unspecified payload to the Authorization Server, and then the Authorization Server replies with a 2.01 (Created) response. In particular, the registration response contains a CBOR map, which includes a "trl" parameter, specifying the path of the TRL resource.

The function 'h(x)' refers to the hash function used to compute the token hashes, as defined in Section 3 of this specifications and according to [RFC6920]. Assuming the usage of CWTs transported in CBOR, 'bstr.t1' and 'bstr.t2' denote the byte-string representations of the token hashes for the Access Tokens t1 and t2, respectively.

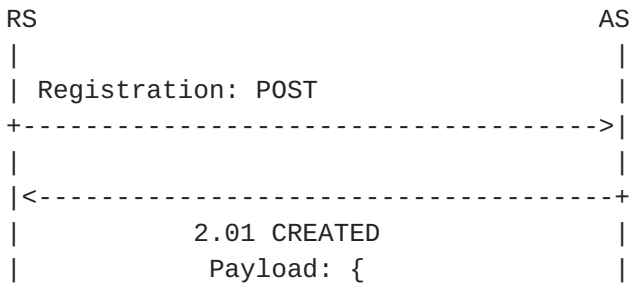

Figure 2: Communication example

9. Security Considerations

Security considerations are inherited from the ACE framework for Authentication and Authorization [I-D.ietf-ace-oauth-authz], from [RFC8392] as to the usage of CWTs, from [RFC7519] as to the usage of JWTs, from [RFC7641] as to the usage of CoAP Observe, and from [RFC6920] with regards to resource naming through hashes. The following considerations also apply.

The Authorization Server MUST ensure that each registered device can access and retrieve only its pertaining portion of the TRL. To this end, the Authorization Server can perform the required filtering based on the authenticated identity of the registered device, i.e. a (non-public) identifier that the Authorization Server can securely relate to the registered device and the secure session they use to communicate.

Disclosing any information about revoked Access Tokens to entities other than the intended registered devices may result in privacy concerns. Therefore, the Authorization Server MUST ensure that, other than registered devices accessing their own pertaining portion of the TRL, only authorized and authenticated administrators can instead retrieve the full TRL. To this end, the Authorization Server may rely on an access control list or similar.

If a registered device has many non-expired Access Tokens associated to it that are revoked, the pertaining portion of the TRL could grow to a size bigger than what the registered device is prepared to handle upon reception, especially if relying on a full query of the TRL resource (see Section 5.1). This could be exploited by attackers to negatively affect the behaviour of a registered device. Short expiration times could help reduce the size of a TRL, but an Authorization Server SHOULD take measures to limit this size.

Most of the communication about revoked Access Tokens presented in this specification relies on CoAP Observe Notifications sent from the Authorization Server to a registered device. The suppression of those notifications by an external attacker that has access to the network would prevent registered devices from ever knowing that their pertaining Access Tokens have been revoked. To avoid this, a registered device SHOULD NOT rely solely on the CoAP Observe notifications. In particular, a registered device SHOULD also regularly poll the Authorization Server for the most current information about revoked Access Tokens, by sending GET requests to the TRL endpoint according to an application policy.

10. IANA Considerations

This document has no actions for IANA.

11. References

11.1. Normative References

[I-D.ietf-ace-oauth-authz]
Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth)", draft-ietf-ace-oauth-authz-33 (work in progress), February 2020.

[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>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/info/rfc6749>.

[RFC6920] Farrell, S., Kutscher, D., Dannewitz, C., Ohlman, B., Keranen, A., and P. Hallam-Baker, "Naming Things with Hashes", RFC 6920, DOI 10.17487/RFC6920, April 2013, <https://www.rfc-editor.org/info/rfc6920>.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <https://www.rfc-editor.org/info/rfc7049>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

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

[RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", RFC 7641, DOI 10.17487/RFC7641, September 2015, <https://www.rfc-editor.org/info/rfc7641>.

[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>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC8392] Jones, M., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "CBOR Web Token (CWT)", RFC 8392, DOI 10.17487/RFC8392, May 2018, <https://www.rfc-editor.org/info/rfc8392>.

11.2. Informative References

[RFC7009] Lodderstedt, T., Ed., Dronia, S., and M. Scurtescu, "OAuth 2.0 Token Revocation", RFC 7009, DOI 10.17487/RFC7009, August 2013, <https://www.rfc-editor.org/info/rfc7009>.

Acknowledgments

The authors sincerely thank Jim Schaad, Goeran Selander and Travis Spencer for their comments and feedback.

The work on this document has been partly supported by VINNOVA and the Celtic-Next project CRITISEC.

Authors' Addresses

Marco Tiloca
RISE AB
Isafjordsgatan 22
Kista SE-16440 Stockholm
Sweden

Email: marco.tiloca@ri.se

Ludwig Seitz
Combitech
Djaeknegatan 31
Malmoe SE-21135 Malmoe
Sweden

Email: ludwig.seitz@combitech.se

Francesca Palombini
Ericsson AB
Torshamnsgatan 23
Kista SE-16440 Stockholm
Sweden

Email: francesca.palombini@ericsson.com

Sebastian Echeverria
CMU SEI
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
United States of America

Email: secheverria@sei.cmu.edu

Grace Lewis
CMU SEI
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
United States of America

Email: glewis@sei.cmu.edu