Notification of Revoked Access Tokens in the ACE Framework

draft-tiloca-ace-revoked-tokens-notification-02

Marco Tiloca, RISE
Ludwig Seitz, Combitech
Francesca Palombini, Ericsson
Sebastian Echeverria, CMU SEI
Grace Lewis, CMU SEI

IETF 108, ACE WG, July 29th, 2020
Recap

› An Access Token may be revoked, before expiration
  – Client or RS has been compromised, or decommissioned
  – Changed access policies
  – Changed ACE profile to use

› New interface at the Authorization Server (AS)
  – The AS maintains one Token Revocation List (TRL) resource
  – The TRL contains the hashes of revoked, not-yet-expired tokens
  – C/RS can GET or GET-Observe from the TRL
  – C/RS retrieve only their own pertaining portion of the TRL

› Benefits
  – Complement token introspection at the AS
  – No need for new endpoints at C or RS
Rationale

› Token hashes computed as per RFC 6920 (binary format)

› TRL resource at the AS
  – CBOR array of Token hashes
  – Add token hashes when Tokens are revoked
  – Remove token hashes when revoked Tokens expire

› Interaction
  – C and RS get the URL to the TRL endpoint upon registration
  – C and RS obtain only hashes of their own pertaining Tokens
  – A registered Administrator gets all Token hashes in the TRL

› Two modes of operations
  – **Full query**: get all pertaining token hashes in the TRL
  – **Diff query**: get the N most recent, pertaining updates to the TRL
Updates from -01

 › Especially addressing
   – Comments from Ben at the June interim – Thanks!

 › Clarified how token hashes are computed
   – Consider what in ‘access_token’ of the response from /token
   – Covered the token transport on both CBOR and JSON
   – Added supporting examples

 › Diff-query mode
   – Simpler interface: GET coaps://example.as.com/revoke/trl?diff=3
   – Simpler format of payload response, with arrays rather than maps

Updates from -01

› Used hash algorithm
  – Now added in the registration response from the AS
  – SHA-256 is mandatory to implement in RFC 6920

› New Appendix A
  – Diff-query mode as an example of the Series Transfer Pattern (STP)
  – draft-bormann-t2trg-stp-03

› Ben’s input for an improved diff-query mode
  – Rather than the N most recent TRL updates ...
  – Get N updates “from where we stopped last time”
  – Revert to full-query if not possible, e.g. information loss/removal at the AS
  – This might actually be a third mode of its own
Updates from -01

› New Appendix B
  – Builds on the “Cursor” pattern of the STP
  – Describes how to achieve the mode suggested by Ben

› Both (a) full-query and (b) diff-query return also a cursor
  – (a) Pointer to the most recent, pertaining TRL update
  – (b) Pointer to the most recent TRL update in the response

› In the enhanced diff-query mode
  – A follow-up request may resume from after the cursor
  – Adjacent batches of TRL updates are possible, limiting excessive latencies

› Handled corner cases
  – No updates, or no updates after the cursor
  – Requested updates have been deleted as too old
Summary and next steps

› Notification of revoked Access Token
  — GET or GET-Observable; full query and diff query
  — Complement token introspection at the AS
  — No need for new endpoints on Clients and Resource Servers

› Version -02 incorporates:
  — Latest review from Carsten and comments from Ben on -01
  — Previous review from Travis and comments from Jim on -00

› Next steps
  — (Third) query mode using the Series Transfer Pattern in the body
  — More workflow examples, e.g. for diff query interactions

› Ready for adoption call (?)
Thank you!

Comments/questions?

https://gitlab.com/crimson84/draft-tiloca-ace-revoked-token-notification
Backup
Protocol overview

Authorization Server

revoke/tr1  TRL: {th1,th2,th3}

th1,th2,th3  th1,th2  th1  th3  th2,th3
v  v  v  v  v

Administrator  Client 1  Resource Server 1  Client 2  Resource Server 2

:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
:: :: :: :: ::
Two types of TRL queries

› Common features
  – Limited to the portion of the TRL pertaining the requester
  – TRL filtering based on authenticated identity of the requester (secure session)

› Full query – GET [Observe: 0] coaps://example.as.com/revoke/trl
  – Request for all pertaining token hashes in the TRL
  – Return a CBOR array, with the Token hashes as elements

› Diff query – GET [Observe: 0] coaps://example.as.com/revoke/trl?diff=3
  – Request for the latest N updates to the pertaining portion of the TRL list
  – Build N entries as CBOR arrays. Each entry refers to an update and has:
    › An element “deleted”, with a CBOR array of Token hashes as element.
    › An element “added”, with a CBOR array of Token hashes as element.
  – Return a CBOR array with the N arrays as element, in reverse chronological order
Example

RS

Registration: POST

AS

<--------

2.01 CREATED
Payload: {
  ...
  "trl" = "revoke/trl"
  "trl-hash" = "sha-256"
}

GET Observe: 0
  coap://example.as.com/revoke/trl/

<--------

2.05 CONTENT Observe: 1
Payload: []
  .
  .
  .

(Access Tokens t1 and t2 issued and successfully submitted to RS)
  .
  .
  .
Example (ctd.)

- (Access Token t1 expires)

- 2.05 CONTENT Observe: 4
  Payload: [h(bstr.t2)]

- (Access Token t1 is revoked)

- 2.05 CONTENT Observe: 2
  Payload: [h(bstr.t1)]
    - 
    - 

- (Access Token t2 is revoked)

- 2.05 CONTENT Observe: 3
  Payload: [h(bstr.t1),
           h(bstr.t2)]
    - 
    - 