TRAM T. Reddy
Internet-Draft S. Nandakumar
Intended status: Standards Track D. Wing
Expires: March 19, 2016 Cisco
B. Williams
Akamai, Inc.
September 16, 2015

Metadata discovery for third party authorized TURN session
draft-reddy-tram-token-metadata-01

Abstract

The operator of the TURN server might want to have fine grained control on the clients usage of the server resources for providing features such as limiting the bandwidth usage, number of allocations and so on. This document proposes a generic mechanism for the operator to introspect the access token to retrieve any policy restrictions imposed by the authorization server on the TURN server resources assigned to the client.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The TURN protocol [RFC5766] is used to setup a relay service (via a TURN Server) to exchange traffic (real time media, data) between peers when direct peer-to-peer connection is not otherwise possible. Due to the costs associated with operating a relay service, it is important to constrain resource usage. For example, the operator might want to limit the number of allocations or bandwidth.

[RFC7635] allows clients to obtain OAuth2.0 access token (of type 'Assertion') authorized by a Authorization Server to access a given TURN server. On receiving such a token, the TURN server validates the token to grant or reject access to the session resources. However, having a token doesn't provide any control for the operator of the TURN server restrict the server's resources. This specification proposes using the mechanism defined in [I-D.ietf-oauth-introspection] to query OAuth2.0 authorization server to determine resource restrictions for this token.

The rest of the document is organized as follows. Section 3 provides procedure for querying the OAuth2.0 Introspection Endpoint and

Section 4 shows the introspection response with the parameters identifying the policy controls associated with the access token.

2. 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].

This document defines the following terms:

Access Token: OAuth 2.0 access token.

Token Introspection: The act of inquiring about the current state of an OAuth 2.0 token through use of the network protocol defined in this document.

Introspection Endpoint: The OAuth 2.0 endpoint through which the token introspection operation is accomplished. The Introspection Endpoint could be a WebRTC server.

3. Introspection Request

For introspecting the meta-information associated with the access token, the TURN server shall execute the procedures defined in Section 2.1 of [I-D.ietf-oauth-introspection].

```
POST {scheme}://{host}:{port}/.well-known/introspection
Accept: application/json
Content-Type: application/x-www-form-urlencoded

{
    "token" : "string"
    "token_type_hint" : "string"
}
```

token REQUIRED. This parameter is defined in [I-D.ietf-oauth-introspection]. The access token is conveyed by the TURN client to the TURN server as discussed in Section 3.1 of [RFC7635].

token_type_hint OPTIONAL. This parameter is defined in [I-D.ietf-oauth-introspection]. The token type MUST be set to 'access_token' defined in [RFC7009]. If the token type is not 'access_token', the server rejects the request with a 400 (Bad Request) error.

Following is a non-normative example request showcasing the introspection request for a given access token.

```
POST /introspect HTTP/1.1
Host: server.example.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded

{

   "token" : "2YotnFZFEjr1zCsicMWpAA"
   "token_type_hint" : "access_token"

}
```

4. Introspection Response

The OAuth2.0 Introspection Endpoint on recognizing the token, responds with a JSON object [RFC7159] in "application/json" format with the following members.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
"active" : "boolean",
"scope" : "string",
"max_upstream_bandwidth" : "unsigned integer",
"max_downstream_bandwidth" : "unsigned integer",
"max_allocations" : "unsigned integer",
"lifetime" : "unsigned integer",
}
```

active REQUIRED. This parameter is defined in [I-D.ietf-oauth-introspection].

scope OPTIONAL. This parameter is defined in [I-D.ietf-oauth-introspection]. For this specification, the scope MUST be ’stun’.

max_upstream_bandwidth REQUIRED. The value of this parameter is an 64 bit unsigned integer that represents the maximum upstream bandwidth permitted for the token in kilobits per second (1 kilobit = 1024 bits).

max_downstream bandwidth REQUIRED. The value of this parameter is an 64 bit unsigned integer that represents the maximum downstream bandwidth permitted for the token in kilobits per second (1 kilobit = 1024 bits).

max_allocations: REQUIRED. 16 bit unsigned integer defining maximum number of allocations that is allowable for the given access token.

lifetime: REQUIRED: The lifetime of the access token, in seconds.

NOTE: Future specifications are allowed to define further top-level members as mandated by the use-cases.

Following is a non-normative example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
"active" : true,
"scope" : "stun",
"upstream-bandwidth" : 4096,
"downstream-bandwidth" : 4096,
"max-allocations" : 1,
}
```

5. INTROSPECTION_ENDPOINT Attribute

This attribute is used by the TURN client to inform the TURN server the FQDN of Introspection Endpoint.

The TURN server establishes an HTTPS connection with the indicated server and sends the above-described communications to that server. The INTROSPECTION_ENDPOINT attribute is a comprehension-optional attribute (see Section 15 from [RFC5389]).

TBD: An alternate approach is to convey the FQDN in the token itself.

6. Notifications from Introspection Endpoint

Introspection Endpoint can send unsolicited responses to notify updates to the metadata associated with the token to the TURN server using HTTP/2 server push mechanism. Examples where such notifications are desired are:

- o The Introspection Endpoint can signal the TURN server to revoke the access token after the call is terminated by setting lifetime to zero.

o When the call switches from audio to video, the Introspection Endpoint notifies the increased bandwidth to the TURN server.

7. Example usage with WebRTC

Below diagram shows a flow where a WebRTC client uses the procedures discussed in [RFC7635] to obtain a OAuth 2.0 access token from the WebRTC server. The TURN Server queries the Introspection Endpoint to determine the metadata associated with the token. Steps 7, 8 and 9 are done using the procedures mentioned in this document.

```
   +-------------------+                         +--------+  +---------------+
   | .........  TURN   |                         |  TURN  |  |  WebRTC Server|
   | .WebRTC .  Client |                         |        |  |               |
   | .Client .         |                         | Server |  |(Introspection |
   | .........         |                         |        |  |  Endpoint)    |
   +-------------------+                         +--------+  +---------------+
     |       |         (1)Allocate request                |                  |
     |       |----------------------------------------->|                  |
     |       |                                          |                  |
     |       |         (2)Allocate error response       |                  |
     |       |         (401 Unauthorized)               |                  |
     |       |<-----------------------------------------|                  |
     |       |         THIRD-PARTY-AUTHORIZATION        |                  |
     |       |                                          |                  |
     |       |                                          |                  |
     |       |      (3)HTTP Request for token           |                  |
     |------------------------------------------------------------------->|
     |       |(4)  HTTP Response with token parameters  |                  |
     |<-------------------------------------------------------------------|
     |OAuth 2.0                                         |                  |
      Attributes                                        |                  |
     |------>|                                          |                  |
     |       |    (5)Allocate request ACCESS-TOKEN      |                  |
     |       |----------------------------------------->|                  |
     |       |                                          |                  |
     |       |      (6)Allocate success response        |                  |
     |       |<-----------------------------------------|                  |
     |       |                                          |(7)Introspection  |
     |       |                                          |  Request         |
     |       |                                          |----------------->|
     |       |                                          |                  |
     |       |                                          |(8)Introspection  |
     |       |                                          | response         |
     |       |                                          |<-----------------|
     |       |<---------------------------------------->|                  |
     |       |             TURN Messages                |                  |
     |       |      ////// integrity protected //////   |                  |
     |       |      ////// integrity protected //////   |                  |
     |       |                                          | (9)Introspection |
     |       |                                          | response         |
     |       |                                          |<-----------------|
```

Figure 1: Metadata discovery for TURN session

8. Alternate Approach

An alternate approach considered by the authors makes use of the access token itself to deliver metadata related to the TURN authentication request. Standard STUN TLV encoded attributes are used to communicate additional metadata associated with the token. Such attributes can be used to define the maximum bandwidth utilization allowed for allocations associated with the token, the maximum number of distinct concurrent allocations, etc.

To include STUN attributes within the body of the access token, the authorization server simply appends them to the access token's plaintext immediately after the lifetime field. The variable length list of attributes MUST consume all of the additional plaintext data space within the body of the access token. No explicit option length value is required or provided.

In order for inclusion of attributes within the plaintext to work correctly in the absence of an explicit length field, one of two things must be true: either the receiver must be able to reliably determine the correct content length from the output of the decryption operation, or the receiver must be able to reliably differentiate between padding bytes and data bytes within the token. AES-128-GCM is an example of the first case, and AEAD mode AES-CBC-HMAC-SHA256 is an example of the second case.

Before parsing the optional data within the access token, the TURN server MUST first perform all token validation required by [RFC7635]. If any of the specified validation checks fail, the TURN server MUST NOT attempt to parse optional attributes.

To interpret the optional attributes within an access token, the TURN server first calculates the amount of option space included in the plaintext by subtracting the size of the base payload data (14 bytes + key_length) from the total payload size. It then interprets the data in the option space as STUN TLV formatted attributes. While parsing the option space, the TURN server MUST apply the same validations to the access token's attributes that it would have applied if the attributes had been included in the outer STUN header (e.g. Verify the data format and value types). If any such validation checks fail, the TURN server MUST reject the STUN request with an error response 401 (Unauthorized).

The following STUN attributes are defined by this document for inclusion in the access token: TBD. Additional attributes may be defined for this purpose in future specifications.

The primary benefits of this method of metadata distribution are:

- It does not impose additional requirements on the Introspection Endpoint for out of band communication with the TURN server.

- Communicating with the Introspection Endpoint may increase the latency associated with TURN allocation request handling.

The primary shortcomings of this method of metadata distribution are:

- Needs a larger TURN packet to accommodate the token. For example, inclusion of the four fields defined above (max_upstream_bandwidth, max_downstream_bandwidth, max_allocations, and lifetime) would increase the token size by around 38 bytes, depending upon whether the AEAD algorithm requires padding.

- The Introspection Endpoint cannot notify the TURN server of changes to the metadata associated with the token.

[NOTE: Backward compatibility with [RFC7635] requires discussion, but it should not be a major issue if the dynamic option space calculation method described is considered acceptable.]

[NOTE: The authors are seeking feedback from the working group on the relative merits of this approach versus the "Introspection Endpoint" approach. Which should we attempt to move forward? Or does each one have enough merit that we should try to advance both?]

9. Security Considerations

The Security Considerations and Privacy Considerations of [I-D.ietf-oauth-introspection] apply to this document.

10. IANA Considerations

10.1. JSON Web Token Claims

This specification requests IANA to register the following values into the IANA JSON Web Token Claims registry for JWT Claim Names.

- Claim Name: "max_upstream_bandwidth"
- Claim Description: Maximum limit of upstream bandwidth
- Change Controller: IESG
- Specification Document(s): Section 4 of [[this document]].

- Claim Name: "max_downstream_bandwidth"
- Claim Description: Maximum limit of downstream bandwidth
- Change Controller: IESG
- Specification Document(s): Section 4 of [[this document]].

- o Claim Name: "max_allocations"
- o Claim Description: Maximum number of allocations
- o Change Controller: IESG
- o Specification Document(s): Section 4 of [[this document]].

10.2. Well-Known ’introspection’ URI

This memo registers the ’introspection’ well-known URI in the Well-Known URIs registry as defined by [RFC5785].

URI suffix: introspection

Change controller: IETF

Specification document(s): This document

Related information: None

10.3. STUN attribute

[Paragraphs below in braces should be removed by the RFC Editor upon publication]

[IANA is requested to add the following attributes to the STUN attribute registry [iana-stun], the INTROSPECTION_ENDPOINT attribute requires that IANA allocate a value in the "STUN attributes Registry" from the comprehension-optional range (0x8000-0xBFFF)].

This document defines the INTROSPECTION_ENDPOINT attribute, described in Section 5. IANA has allocated the comprehension-optional codepoint TBD for this attribute.

11. Acknowledgements

Authors would like to thank Ram Mohan for comments and review.

12. References

12.1. Normative References

[I-D.ietf-oauth-introspection]
Richer, J., "OAuth 2.0 Token Introspection", draft-ietf-oauth-introspection-11 (work in progress), July 2015.

[iana-stun]
IANA, , "IANA: STUN Attributes", April 2011, <http://www.iana.org/assignments/stun-parameters/stun-parameters.xml>.

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

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, DOI 10.17487/RFC5389, October 2008, <http://www.rfc-editor.org/info/rfc5389>.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, DOI 10.17487/RFC5766, April 2010, <http://www.rfc-editor.org/info/rfc5766>.

[RFC7635] Reddy, T., Patil, P., Ravindranath, R., and J. Uberti, "Session Traversal Utilities for NAT (STUN) Extension for Third-Party Authorization", RFC 7635, DOI 10.17487/RFC7635, August 2015, <http://www.rfc-editor.org/info/rfc7635>.

12.2. Informative References

[RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, DOI 10.17487/RFC5785, April 2010, <http://www.rfc-editor.org/info/rfc5785>.

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

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <http://www.rfc-editor.org/info/rfc7159>.

Authors' Addresses

Tirumaleswar Reddy
Cisco Systems, Inc.

Email: tireddy@cisco.com

Suhas Nandakumar
Cisco Systems, Inc.

Email: snandaku@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Brandon Williams
Akamai, Inc.
8 Cambridge Center
Cambridge, MA 02142
USA

Email: brandon.williams@akamai.com