ACE Working Group
Internet-Draft
Intended Status: Standards Track
Expires: October 14, 2016
April 12, 2016

L. Seitz
SICS Swedish ICT

Transporing access tokens in session resumption tickets
draft-seitz-ace-ticket-token-transfer-00 (TTT)

Abstract

This memo presents a method of transferring an access token from a client to a resource server in a (D)TLS handshake, based on Session Resumption without Server-Side State (RFC 5077).

Status of this Memo

Copyright and License Notice

Table of Contents

1. Introduction

The default way of transferring an OAuth access token to a resource server (RS) via CoAP is defined in [I-D.ietf-ace-oauth-authz] as POSTing to a well-known resource, namely /authz-info on the RS. This solution might not be ideal in all cases, as it requires an extra message exchange and the RS needs to perform a lookup when the request arrives to determine which token matches this request. Therefore this memo describes how to transfer an access token inside a server state ticket used for (D)TLS session resumption without server state [RFC5077].

1.1 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 [RFC2119]. These words may also appear in this document in lowercase, absent their normative meanings.

2. Ticket format

The StatePlaintext structure from section 4 of RFC 5077 is modified as follows:

```
      struct {
          ProtocolVersion protocol_version;
          CipherSuite cipher_suite;
          CompressionMethod compression_method;
          opaque master_secret[48];
          ClientIdentity client_identity;
          uint32 timestamp;
          uint16 access_token_length;
          opaque access_token;
      } StatePlaintext;
```

Where the access_token field contains a representation of the access token, readable for the RS, and the access_token_length field gives the length of this token in bytes.

Furthermore the following changes are made to ClientAuthenticationType structure in order to support raw public keys (RPK):

```
     enum {
         anonymous(0),
         certificate_based(1),
         psk(2),
```

```
            rpk(3)
        } ClientAuthenticationType;
```

Finally the ClientIdentity structure is modified as follows, also to support RPK:

```
     struct {
        ClientAuthenticationType client_authentication_type;
        select (ClientAuthenticationType) {
           case anonymous: struct {};
           case certificate_based:
              ASN.1Cert certificate_list<0..2^24-1>;
           case psk:
              opaque psk_identity<0..2^16-1>;   /* from [RFC4279] */
           case rpk:
              opaque ASN.1_subjectPublicKeyInfo<1..2^24-1>;
              /* from [RFC7250] */
        };
     } ClientIdentity;
```

All other parts of RFC 5077 remain unchanged. The RS MUST process the ticket as specified in RFC 5077, and additionally it MUST verify the validity of the token contained in the access_token field, and if it is valid it MUST store it for future use. The validity check MUST include a check of the token binding to the client identity given in the client_identity field. If the token is not a bearer token, the RS MUST reject a ClientAuthenticationType of anonymous and abort the handshake with an illegal_parameter error.

3. Security Considerations

All security considerations from RFC 5077 apply equally to this memo. Furthermore the methods for verifying the validity of an access token may vary widely depending on the token type. Implementers should carefully consider how to avoid mixing up different token types (e.g. bearer tokens vs proof-of-possession tokens) which require different verification methods. Resource Servers MUST NOT accept a session resumption ticket containing a token for which the RS can not determine the validity (e.g. because it cannot interpret the token format).

4. Privacy Considerations

The privacy considerations from RFC 5077 apply equally to this memo. The length of the ticket might leak information about the fact that it contains a access token, and possibly about the format and

contents of the token. Adversaries having good knowledge of the different possible access tokens in a specific application, could determine which kind of access the token authorizes based on its length. If such attacks are of concern, a padding method for the token should be considered.

5. IANA Considerations

This document does not require any actions or assignments from IANA.

6. Acknowledgements

Jim Schaad originally suggested this approach, and Hannes Tschofenig explained the rudimentary details at IETF 95.

7. References

10.1 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, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, DOI 10.17487/RFC5077, January 2008, <http://www.rfc-editor.org/info/rfc5077>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <http://www.rfc-editor.org/info/rfc6347>.

[RFC7250] Wouters, P., Ed., Tschofenig, H., Ed., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", RFC 7250, DOI 10.17487/RFC7250, June 2014, <http://www.rfc-editor.org/info/rfc7250>.

10.2 Informative References

Author's Address

Ludwig Seitz

SICS Swedish ICT AB
Scheelevagen 17
22370 Lund
SWEDEN
EMail: ludwig@sics.se