ACE Working Group
Internet-Draft
Intended status: Informational
Expires: September 14, 2017

J. Zhu
Huawei

March 13, 2017

Offline usage of ACE
draft-zhu-ace-offline-00

Abstract

[I-D.ietf-ace-oauth-authz] defines a framework for both authentication and authorization in constrained Internet of Things (IoT) environments. A constrained node in this framework may have constraints in computational capability, memory storage, lack of user interface, transmission bandwidth and/or power supply. Battery-powered devices are widely used in IoT deployments and they sleep most of their lifetime for battery saving. Hence, they are usually disconnected from other nodes. This draft provides an overview of the disconnection use cases and discusses offline authentication and authorization solutions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 14, 2017.

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 4
3. Cases 4
3.1. Case 1 Client-AS disconnection 5
3.1.1. Sub-case 1 Client instructs the RS to obtain authorization information from AS 5
3.1.2. Sub-case 2 Introspection Aided Token Validation . . . 7
3.1.3. Sub-case 3 RS caches authorization information . . . 7
3.2. RS-AS disconnection 7
3.2.1. Sub-case 1: Local Token Validation 7
3.3. Client-RS disconnection 7
4. Security Considerations 8
5. IANA Considerations 8
6. Acknowledgements 8
7. Changelog 8
8. References 8
8.1. Normative References 8
8.2. Informative References 9
Author's Address 9

1. Introduction

[I-D.ietf-ace-oauth-authz] defines a framework for both authentication and authorization in constrained Internet of Things (IoT) environments. The framework is based on a set of building blocks including OAuth 2.0 and CoAP.
Figure 1/[I-D.ietf-ace-oauth-authz] describes the basic ACE protocol flow. The diagram is repeated below.

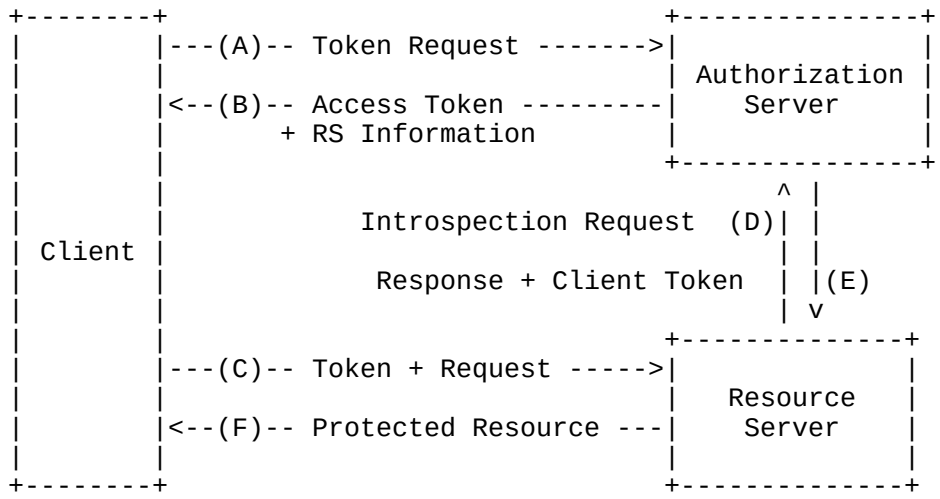

Figure 1: Basic Protocol Flow

(A) The client makes an access token request to the /token endpoint at the Authorization Server (AS).

(B) The AS successfully processes the request from the client, then returns an access token and some RS information.

(C) The client interacts with the resource server (RS) to request access to the protected resource and provides the access token.

(D) The RS may make an introspection request to the /introspect endpoint at the AS to get more information about the access token.

(E) The AS validates the token and returns the most recent parameters associated with it back to the RS.

(F) The RS uses the token information to process the resource access request and returns the protected resources back to the client.

Note: Step D and E are optional steps as the RS can process the access token information locally depending on the deployment configurations.

There may be many constraints for a constrained IoT device such as limited computational capability, memory storage, lack of user interface, transmission bandwidth and/or power supply. According to the [I-D.ietf-ace-actors], either the client or the RS MAY be a constrained node. One critical issue for IoT ecosystems is that more and more constrained devices are battery-powered, e.g. smart water

meters. These battery-powered constrained devices sleep most of their lifetime to save power. What's more, in deployments the underlying network between different nodes may vary from cellular to WLAN even NFC. That means any two nodes of the ACE framework may be disconnected from each other.

As a result of Figure 1, there are 3 different possible disconnection cases between the nodes in the ACE framework:

1. Client-AS disconnection

2. RS-AS disconnection

3. Client-RS disconnection

This document provides an overview of these cases and discusses offline authentication and authorization solutions based on the ACE framework for each of the cases.

The cases discussed in this document utilise the A to F designations from Figure 1 to maintain a relation to the functional steps.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119].

This specification requires readers to be familiar with all the terms and concepts that are discussed in [I-D.ietf-ace-oauth-authz] and [RFC7252].

3. Cases

This section discusses the disconnection cases including:

1. Client-AS disconnection

2. RS-AS disconnection

3. Client-RS disconnection

Each of the cases may have one or more sub-cases.

For each case there is a brief description at the beginning, and then a possible solution for the disconnection case is discussed.

3.1. Case 1 Client-AS disconnection

In this case we consider the case where the Client is disconnected from the Authorization Server when the Client wants to access a resource on the Resource Server. This usually happens when the network between client and AS goes down, but the client can communicate with the RS via another network.

3.1.1. Sub-case 1 Client instructs the RS to obtain authorization information from AS

This example shows the interaction between a remote controller (Client), a smart television (RS) and a Hub (AS). The remote controller is disconnected from the AS because its WIFI function doesn’t work well. However it can communicate with the smart TV via Bluetooth.

This access procedure involves all the steps shown in Figure 1. In this case, it is assumed that there is a DTLS connection between the client and RS and a separate DTLS connection between the RS and AS.

The client firstly tries to turn on the RS without any authorization information.

C: The client sends a request message to the RS in order to change the state of the switch. However this message does not contain any authorization information.

F: After receiving the request message, the RS verifies it and sends an authorization verification failure response back to the client. The payload of the response MAY contain the AS information in order to instruct the client to obtain an access token from the right address.

Messages C and F is shown in Figure 2.

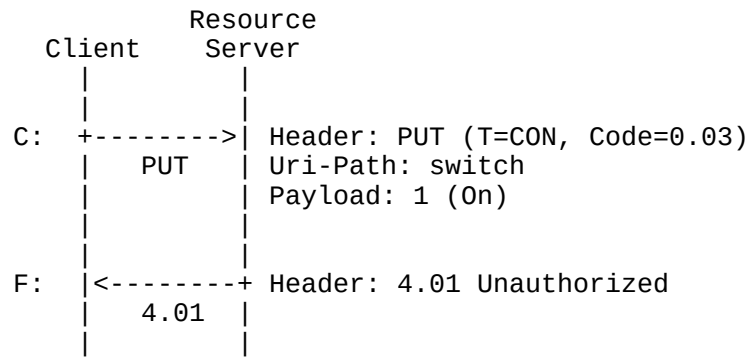

Figure 2: Authorization Failure

After receiving the unauthorized failure message from the RS, the client then tries to request an access token from the AS.

A: The client sends an authorization request to the AS.

B: Because the client is disconnected from the AS, the access token request does not receive a response from the AS and the request times out.

Message A and B are shown in Figure 3.

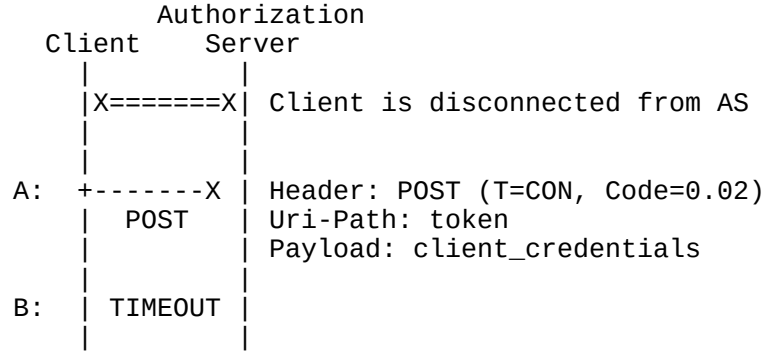

Figure 3: Authorization Timeout

Question: Would it be possible to use the resource server as a proxy to get the authorization information?

3.1.2. Sub-case 2 Introspection Aided Token Validation

In this scenario we consider the same example shown in Section 3.1.1. The difference is that the client has previously (when it could communicate with the AS) received a pre-provisioned long-lived access token before it went offline. The RS uses its online connectivity to validate the access token with the AS.

Note: This is the same use case as the example described in section E.2 of [I-D.ietf-ace-oauth-authz].

3.1.3. Sub-case 3 RS caches authorization information

In this section we consider the same case mentioned in Section 3.1.1.

It is assumed the client can communicate with the AS over a DTLS channel before it goes offline. A DTLS channel is also established between AS and RS as well as a separate channel between the client and RS.

The RS has the capability to cache client authorization information.

Question: Would it be acceptable for the RS to have its cache managed by the client?

3.2. RS-AS disconnection

3.2.1. Sub-case 1: Local Token Validation

In this scenario we consider the case where the resource server is offline, i.e. it is not connected to the AS at the time of the access request. This access procedure involves steps A, B, C, and F of Figure 1.

Since the resource server must be able to verify the access token locally, self-contained access tokens must be used.

Note: This case is the same as the example described in section E.1 of [I-D.ietf-ace-oauth-authz].

3.3. Client-RS disconnection

In this scenario we consider the case where the client is disconnected from resource server at the time of the access request. For example, both a mobile phone (Client) and a thermostat(RS) are connecting to a same cloud server(AS). The phone has no connection to the thermostat, but the AS should provide a mechanism for the

client to query the temperature remotely. This access procedure involves steps A, B, D, and E of Figure 1 as shown below.

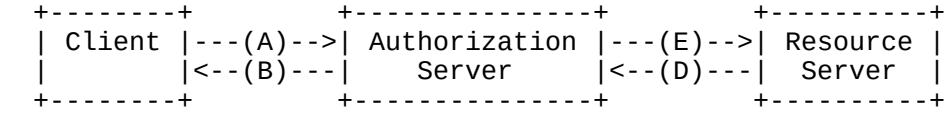

Figure 4: Client-RS disconnection

In this case, it is assumed that a DTLS channel is established between the client & AS, and a separate DTLS connection between the AS & RS as well. The AS SHOULD act as proxy and can forward the resource access request by the client to the RS. The client prior to sending a message to the AS, tried to access the resource directly. However it did not get a successful response due to disconnection between these two nodes. So the client then tries to access the resource via the AS.

Question: Would it be acceptable for the AS to act as a proxy for requests to the RS?

4. Security Considerations

This document addresses authorised access to resources in device disconnection scenarios.

5. IANA Considerations

TBD.

6. Acknowledgements

TBD.

7. Changelog

Initial version

8. References

8.1. Normative References

[I-D.ietf-ace-actors]
Gerdes, S., Seitz, L., Selander, G., and C. Bormann, "An architecture for authorization in constrained environments", draft-ietf-ace-actors-05 (work in progress), March 2017.

[I-D.ietf-ace-oauth-authz]
Seitz, L., Selander, G., Wahlstroem, E., Erdtman, S., and H. Tschofenig, "Authentication and Authorization for Constrained Environments (ACE)", draft-ietf-ace-oauth-authz-05 (work in progress), February 2017.

[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>.

8.2. Informative References

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

Author’s Address

Jintao Zhu
Huawei
P.R.China

Email: jintao.zhu@huawei.com