Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 9, 2019

S. Echeverria
CMU SEI
L. Seitz
RISE
D. Klinedinst
G. Lewis
CMU SEI
March 8, 2019

ACE Clients in Disadvantaged Networks
draft-secheverria-ace-client-disadvantaged-00

Abstract

This document describes a set of recommendations to use when implementing ACE/OAuth 2.0 clients that are working in disadvantaged networks. Issues such as token revocation have a much higher priority in scenarios where Resource Servers are IoT devices, and network connectivity is limited and intermittent.

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 https://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 9, 2019.

Copyright Notice

Table of Contents

1. Introduction 2
2. Sample Scenario 3
3. Recommendations 3
3.1. Use of Client Introspection for Token Revocation 3
3.1.1. Procedure 3
3.1.2. Specific Recommendations 4
3.1.3. Alternatives 5
4. IANA Considerations 6
5. Security Considerations 6
6. Acknowledgements 6
7. Normative References 6
Authors' Addresses 7

1. Introduction

Authentication and authorization in IoT (Internet of Things) devices can be difficult due to constraints in terms of memory, processing, user interface, power and communication bandwidth. OAuth 2.0 and derived standards, such as ACE, can be still applied to these scenarios, often with some modifications. However, when IoT devices are working in disadvantaged networks, there are even greater constraints in terms of communication bandwidth. Nodes in disadvantaged networks operate in what are called DIL environments (disconnected, intermittent, limited), which means that there is limited and unreliable connectivity between nodes with potentially periods of full disconnection. This document will focus on practices that are recommended for clients using ACE/OAuth 2.0 while working with IoT devices in disadvantaged networks.

There are cases in which a client may need to obtain further information about a token without communicating with a Resource Server (RS). One such case is when a client needs to know the active status of a token that it possesses. This is particularly useful in disadvantaged environments where RS impersonation and sabotage are likely threats.

Section 2 describes a sample scenario and Section 3 describes recommendations for client implementation, including the use of client introspection: ensuring only authorized clients can perform client introspection, enabling decryption of self-contained tokens, and limiting information returned in the introspection response.

2. Sample Scenario

A sample scenario is the following: let's assume we have IoT devices that are deployed over a large area to monitor it after an earthquake. These IoT devices may be small sensors of different types (temperature, motion detectors, etc.) that are constantly collecting information from their environment. Each of these IoT devices will act as an RS, and we want to be able to give authorization to access their resources to mobile clients. The Authorization Server (AS) will be mostly static, or slow moving; it could be deployed in a nearby building, or carried along in vehicles if there is no central location. Clients would most likely be smartphones or tablets, carried by users in the field. Due to the mobility of the clients and the large area over which the RSs are deployed, clients would only intermittently have connectivity to both the AS and to each RS. Clients would ask the AS for access tokens when they are in range of the AS, and use the tokens to get information from the RSs when they are in range of the IoT devices.

In this situation, being opportunistic about what to do when a client gets in range of an AS is an important thing to consider. It is also highly likely that clients or RSs may be impersonated or sabotaged. This makes it a high priority to identify tokens associated to a compromised RS or sent to a compromised client.

A specific situation for this would be if the AS admin learns that a certain RS has been compromised. The AS does not have constant connectivity to clients, so it can't let them know right away about the issue. However, it wants to prevent all clients that had tokens to communicate with that RS that they should no longer use those tokens. The AS admin can manually mark all tokens issued to that RS as an audience as revoked (internally). However, a means to let clients know about the revocation of their tokens would be needed.

3. Recommendations

3.1. Use of Client Introspection for Token Revocation

3.1.1. Procedure

One way to let clients know when a token has been revoked is to extend the existing protocol to add specific messages to handle this. But an alternative, simpler way would be to use client introspection. The end goal is to be able to revoke tokens for a RS that has been compromised, by letting clients poll information about the tokens, and then letting them know that they have been revoked.

A client can opportunistically poll an AS using the same introspection mechanism defined in the OAuth 2.0 Token Introspection RFC [RFC7662], to obtain information on whether a specific token is still valid or not. That RFC defines a method for querying an Authorization Server (AS) for metadata about a token. The introspection process focuses on how a Resource Server (RS) could benefit from this information. This is because, in most cases, the token is assumed to be opaque to the client, as it is intended to be a secure way of sending information to a RS, without the client being able to modify it.

This same mechanism can be used to detect revoked tokens opportunistically whenever a client gets in range of an AS. It should work in the following way:

1- A client that gets in range of the AS, uses that opportunity to contact the AS and to ask it about the state of its non-expired tokens. More specifically, it sends a client introspection request for each of the non-expired tokens it is using.

2- The AS replies to the client with information about whether each token is "active" or not. For every token that has been revoked, it returns that the token is not active.

3- The client receives the response and purges non-active tokens from its list of tokens.

Thus the client will be protected from contacting the compromised RS. Of course, this does not prevent the client from contacting the RS before it can access the AS and ask about the tokens, but there is not much that can be done about it until the client is able to communicate with the AS.

3.1.2. Specific Recommendations

The following recommendations are useful to consider when implementing client introspection:

1- The AS should have a way to limit which clients are allowed to send introspection requests. This ensures that only clients that really do need the information are allowed access to it.

2- The "kid" header parameter as defined in [RFC7519] and [RFC8152] should be used when the token is encrypted in a structured information object such as a JSON Web Token (JWT) [RFC7519] or CBOR Web Token (CWT) [RFC8392]. The AS can store a key ID in this header that can be associated with the RS key used during encrypted token creation. If the AS does this when generating every encrypted token,

then it should always be able to decrypt that token on an introspection request coming from a client or from a RS.

This is needed because an encrypted token can only be decrypted if the proper key is known. When an RS performs introspection, the AS can use the identity of the RS as a hint to find the related key. However, if a client is performing the introspection request, the AS receiving the request needs more information to know what audience the encrypted token was issued for in order to decrypt it properly.

3- Only the value of the "active" parameter should be returned for introspection requests coming from clients. An introspection response has several parameters, but all of them are optional except for the "active" parameter. The "active" parameter can be used to indicate that a token has been revoked, and does not provide any information about the claims, which the client should usually not need. This prevents the disclosure of additional information to the client.

3.1.3. Alternatives

An alternative way to handle token revocation would be to prevent the AS from issuing more tokens for the same RS/audience, and for the client to request a new token each time it is in range of the AS. In this case, tokens would not be revoked, but rather clients would be implicitly notified to no longer contact a specific RS. However, this has at least two downsides. First, a client would have to request a new token each time it is in range of an AS, constantly, to be able to detect token revocation by getting the token request to be denied. This could lead to many tokens issued to the same client and for the same RS in a short period of time, which may not be even used in that timeframe. In addition, this generates additional traffic in an already constrained network. Second, client would be interpreting a denial to issue a token from an AS as a warning not to contact that RS anymore. This could lead the client to dump a previous token that it has for that RS, to prevent potentially dangerous contact with it. However, the denial may be for other reasons, but there is no way to differentiate when denying a token request to a client. Thus, a client may end up dumping working tokens because of a potentially different issue with new token generation. In summary, this option depends on the client making too many assumptions to successfuly prevent it from accessing a compromised RS. Using client introspection to detect revoked tokens is a much simpler and direct way of handling this issue.

Another similar alternative to revoking tokens is to issue tokens with very short lifetimes. In this case, even if a device having a token is compromised, the short lifetime will make that token expire

quickly, making revocation notifications unnecessary. The main problem with this option in disadvantaged networks is that clients will not often be in range of the AS that issues the tokens or of the RS they want to use the token with. Thus, if tokens have very short lifetimes, they may not last long enough for a client to actually send that token to the RS it needs to contact. Or even if it does, if the token expires shortly afterwards, the client will not be able to contact that or other RS in the same audience again until it comes in range of the AS to obtain a new token. Thus, in this type of environments, the lifetime of a token must be carefully balanced in relation to its intended use and the frequency the devices will be in range of each other.

4. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

5. Security Considerations

There are some potential security issues with the recommendations described in this document. Because the AS would accept introspection requests from a client, claim information associated to the tokens and not intended for a client could be sent back to it in a response. The recommendations above explicitly indicate to only send the "active" parameter as the response to this type of request, but it is still up to the implementation to do this properly, and to properly identify a device as a client (or more specifically as a device to send limited information to in a reply). If this is properly done, compromised or rogue clients sending introspection requests would not be able to obtain more information than the token active status from these types of introspection requests.

6. Acknowledgements

7. Normative References

[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>.

[RFC7662] Richer, J., Ed., "OAuth 2.0 Token Introspection", RFC 7662, DOI 10.17487/RFC7662, October 2015, <https://www.rfc-editor.org/info/rfc7662>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

[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>.

Authors' Addresses

Sebastian Echeverria
CMU SEI

Ludwig Seitz
RISE

Dan Klinedinst
CMU SEI

Grace Lewis
CMU SEI