Human Rights Considerations in the OAuth 2.0 Framework

•••

Sunil Abraham
Centre for Internet and Society, India

Thanks to support and contributions from

Anand Venkatanarayan [from industry]

Udbhav Tiwari [CIS Programme Manager]

Rohini Lakshané [ex-CIS employee]

Srinivas Narasimhan Cummaragunta [intern from National Law School of India University, Bengaluru]

Pranit Bauva [intern from Indian Institute of Technology, Kharagpur]

Sidharth Ray [intern from School of Law, Christ University, Bengaluru]

Work done? / pending

Done:

1. RFCs and IETF documentation, blogs and news
2. Video recordings of IETF meetings.
3. Academic literature.

Pending:

1. Mailing list archives analysis.
2. Interviews with members of the OAuth working group.
3. Analysis of all academic papers.
4. Review of the consideration with members of PITG and members of OAuth WG.
5. Posts on mailing list and blog.

Standard vs. Framework

1. Are standards easier for HRPC when compared to frameworks? Is this also true for implementation reports.
2. Is the scope of a standard smaller when compared to framework? What is the perimeter of analysis for a framework?
3. Are standards more important than frameworks when it comes to human rights?
4. What should the next standard/framework for an HRIA be?
5. When should the standard be blamed? [For ex. granularity as part of the privacy consideration]

[*] Discounting math-phobia and crypto-phobia

RFC 8280 - Research into Human Rights Protocol Considerations

1. The security section was difficult because of 1. The answer to the first question would be common to all standards 2. Identification of new attacks for a non-expert is not easy and 3. there are many permutations and combinations for different scenarios. For ex. national ID projects using OAuth - who will use what to attack who and with what consequences for human rights.
2. Outcome transparency feels much more like mapping on unintended effects rather than “the right to explanation” in GDPR and AI regulation. Therefore again this is very difficult for a non-expert and perhaps also require time for serendipity.
3. The assumption has to be made that the workgroup has optimized the trade off between the different rights. For ex. security vs. accessibility.

Connectivity

- OAuth 2.0 does not add any application-specific functionality to intermediary nodes that might not already exist between the end nodes of the OAuth framework.
- The framework cannot be developed in a stateless manner, due to nature of transactions that occur on it.
- Optimization remains an issue for low bandwidth and high latency connections, especially in developing countries.

Privacy

- OAuth 2.0 RFC does not mandate, require or prescribe any maximum authentication period for the grant of access by the resource owner.
- Grant can only be revoked absolutely, with either a positive or negative action (of granting or revoking access) without the access being time bound in any manner.
- Solutions - Periodic reminder of the applications to which access has been granted by a resource owner or a maximum time limit after which the access grant automatically expires.

Content Agnosticism

- OAuth protocol is content agnostic from the perspective of different MIME-types, encoding standards, encryption standards and file formats.
- The content delivered from the content server to the end user is outside the scope of the standards.

Security

- Three major security oriented RFCs on oAuth already exist: e OAuth 2.0 Authorization Framework, The OAuth 2.0 Threat Model and Security Considerations and OAuth Security Topics
- A number of security concerns exist with the framework, allowing parties access to access and refresh tokens, from which the data within the scope of the token can be extracted.
- In one study, data showed that 59.7% of oAuth implementations were faulty and vulnerable to attacks, both from a design choice and implementation perspective.

Internationalization

- Most of OAuth is compliant with internationalisation and localisation support.
- Few exceptions are in error codes (both action, response and description) which choose to stick to ASCII.
- The effect that this has is that a user cannot implement an OAuth framework entirely in a language that doesn’t support ASCII
- Two possible solutions: Make UTF-8 mandatory throughout the standard and appendix or leave ASCII for the error request and token while making UTF-8 mandatory for the error descriptions.

Censorship Resistance

- If the authorization server is blocked in a particular jurisdiction, then content servers are harder for the resource owner to access, if not impossible.
- For example, Twitter and Facebook being blocked in China impacts the use of certain web apps that only support the Twitter OAuth service.
- Some workarounds have been proposed for this issue, for example by J Barends, but they are largely implementation dependant, with nothing present in the protocol that recommends or asks for it in particular.

Open Standards

- The OAuth framework is completely documented from an implementation perspective.
- It does not implicate any patents and there is no dependence on proprietary code.
- The protocol does not favour any proprietary software implementation.
- The implementation of the framework does not depend on any other standard that is not available on a royalty-free basis.

Heterogeneity

- The OAuth framework has heterogeneity support by design because of its limited scope and its wide real world use cases in diverse hardware and software configurations.
- Work to support OAuth on other application protocols XMPP, MQTT and CoAP has started in the past few years as well.
- However, the strong heterogeneity also leads to difficulty in conducting human rights impact assessments, especially due to lack of clarity and data on implementation.

Anonymity and Pseudonymity

- The anonymity criterion in the human rights protocol considerations is in conflict with raison d'être of the OAuth standard.
- A proposed standard called Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants [RFC7521] envisaged a scenario wherein a client acts on behalf of an Anonymous User but there hasn’t been much traction on it inside the IETF.
- The framework does have any specific requirement or recommendation that explicitly enables or prevents pseudonymity. The pseudonymity of an individual and their data is entirely dependent on the authorization server of the platform being used, as well as state regulation for the same.

Reliability

- The OAuth framework consists of 5 steps that are performed in a particular order: user authorization request --> user authorizes application --> authorization code grant --> access token request --> access token grant.
- The OAuth 2.0 framework is largely fault tolerant, which allows it to heal partially or recover from a fault. For example, in the 5 steps above, any interruptions due to network failures between steps 3 to 5 can be corrected/continued from the last successful step.
- Additionally, the OAuth framework does not degrade, either gracefully, or maliciously. This is because the framework either works or it doesn’t.

Confidentiality

- As per the protocol, none of the steps involved are required (or not required) to be encrypted. Therefore the contents of any token that is intercepted or illegally acquired, are openly available to the person in possession of the token.
- OAuth does not protect the confidentiality of the data or identifiers.
- In some implementations, however, some level of confidentiality is maintained by implementing encryption through HTTPS or through implementation of the JWT extensions.

Integrity

- As per the protocol, none of these steps are required to be encrypted, as shown in the “Confidentiality” section.
- Additionally, the protocol does not mandate the signing of tokens, which means that there is no definite way to ensure the integrity of the data, especially if it has been manipulated.
- JSON Web Token implementation in OAuth accounts for signing but only for JWT implementations, which are a sub-set of the diverse use cases and implementations for OAuth.

Adaptability

- Since OAuth 2.0's inception in 2010, a number of extension have been developed upon the OAuth framework. Examples include: OAuth 2.0 Device Flow [May 31, 2017], OAuth 2.0 Token Introspection [Oct 2015], PKCE [Sep 2015], Native Apps [June 9, 2017], JSON Web Token [May 2015], OAuth Assertions Framework [May 2015], SAML2 Bearer Assertion [May 2015], JWT Bearer Assertion [May 2015].
- Furthermore, protocols have enn entirely developed upon the existing OAuth framework, including examples like OpenID Connect, Green Button and the User-Managed Access (UMA) by the Kantara Initiative.

Outcome Transparency

- Due to its nature as a framework and not a protocol or standard, outcome transparency feels much more like mapping on unintended effects rather than “the right to explanation” in GDPR and AI regulation.
- Therefore again this is very difficult for a non technical expert to do and perhaps also call for more time for serendipity to work in real world use cases.

Inapplicable/Irrelevant/Undone Considerations

- Localization - Same as and based on Internationalisation
- Decentralization - Completely dependent on implementation? Assessment of native client vs. IoT.
- Accessibility - Not protocol dependent, completely based on implementation of client side implementation.
- Authenticity - Same as Confidentiality and Integrity, both of which are heavily based on signing and encryption, which are not available in the framework.
- Outcome transparency - Undone.

•••

sunil@cis-india.org/+91 9611100817/@sunil_abraham