OAuth WG Meeting Minutes

Meeting Minute Taker: Hannes Tschofenig

Monday's Agenda

1. Chairs update - Rifaat/Hannes (15 min)

Status of work items - RFC 9207 (OAUTH 2.0) published, JWT Response in RFC Ed queue, three other documents (JWK Thumbprint URI, Rich Authorization Requests, Security Best Current Practice) have WG consensus but are pending write-ups by the chairs.

2. DPoP - Mike/Brian (45 min)

https://datatracker.ietf.org/doc/draft-ietf-oauth-dpop/

Brian goes through his slides describing the DPoP mechanism

(Small bug in slide #12 where the example is missing a WWW-Authenticate: DPOP indication)

Draft -04 added the option for a server-provided nonce, draft -05 and -06 refined and clarified this mechanism

Next steps:

- Is the media type registration necessary?
- Six authors in the draft
- WGLC (?)

Roman: If the group believes that all six authors are important, then Roman will defend it.

Mike: JWT type claim is intended as a media type with a special syntax. We should just registry it, even if it is not used.
Brian: That's correct. Maybe we should just do it.
Mike: I will create a PR.
Brian: This will be necessary.

Suhas Nandakumar: I supporting the WGLC. We (at Cisco) are planning to use the specification and have been following it for a while.

Rifaat: Update the document and we will start a WGLC.

3. Redirection Attacks - Rifaat (30 min)

https://mailarchive.ietf.org/arch/msg/oauth/4-YCJzeDH4NH-ge9OF8bAbqWgIE/

Rifaat is going through his slides.

Rifaat asks the group where the error handling should be rethought. Where should the decisions / recommendations be captured?

Justin: This is the known trade-off of doing any communication on the front channel. The updated BCP talks about it, at a high level (not the specifics of the threats). To me it makes no sense to document the issue in a separate draft. This is not a new attack.

Mike: Adding descriptions of what we already know is a good way. I don't think we should be making normative changes to our specifications. We should ask implementers to think about doing the redirect back under certain situations.

Daniel: I agree that the security BCP is the right place for it, put maybe the next version of the security BCP.

Rifaat: why do you want to wait for the next version?

Daniel: There are typically are a lot of discussions. It will delay the security BCP.

Rifaat: It is a well-known issue.

Daniel: If it happens quickly, then we should do it.

Mike: We still have to get through IESG and IETF review. There is a lot of opportunity for adding the text. Those who care should find a room and write a few paragraphs.

Rifaat: We have side-meetings and could discuss this topic.

4. OAuth 2.1 - Aaron (30 min)

https://datatracker.ietf.org/doc/draft-ietf-oauth-v2-1/

Aaron gives an update of OAuth 2.1.

The specification now makes explicit statments about the OAuth implicit flow. It still allows OpenID Connect to define its own implicit flows, such as response_type=id_token.

Justin congratulates Aaron for the work. #97 - the notion of what is backwards compatibility will vary (depending on whether you are talking about a client, AS or RS). Good that this is clarified. Keep in mind what it means to all the parties in the ecosystem. Some breaking changes are OK.

issue #46 - iss response parameter

RFC 9207 is now available and we can use iss response parameter to mitigate against AS mixup attacks. It only applies to clients that talk to multiple ASs. Is the iss response parameter now best current practice, or should we reference it otherwise?

Mike: I was one of the persons who said “wait”. I am fine folding in guidance for clients talking to multiple ASs to prevent mix-up attacks.

Aaron: Should we reference or copy text?

Mike: Include an example of using it and I would be fine referencing it. It is a judgment call.

Daniel: I would support Mike. Very brief guidance, a most common case and guidance when you don’t have the concept of an issuer. There are some corner cases in the RFC. It makes sense to describe the corner cases.

Brian: The actual mechanism is something that the AS has to do. How do you describe this in the document when the AS has to do something but the client knows that it talks to multiple ASs? The AS has to be build for in order for the client to use it.

Aaron: Can the client also use multiple redirect URIs?

Daniel: We discussed this in the security BCP to defend in the security BCP but we also say when it is not a good idea. Only the iss response parameter is a robust defense.

Aaron: Realistically it is not really a problem to describe the conditions.

Mike: The iss response parameter was motivated by OpenID Connect Issuer. If you are doing pure OAuth you don’t have an issuer reponse. In the ID Token there is always this parameter.

Vittorio: You want to have the issuer in the message header without having to dig into the ID token.

Daniel: In pure OAuth there is no issuer. I am in favour of using OAuth meta-data.

Fillip: One of the reasons is to make a decision before relying in OpenID Connect.

Aaron: We more or less agree on the issue.

issue #101: Access tokens must have a limited lifetime.

Many deployments do not, however, follow this guidance. It feels strange to publish another document that says the same message again. Maybe this recommendation in RFC 6750 does not match deployment reality.

Mike: I inherited text from another Aaron and I had no intent to add a MUST into the security consideration section. There is nothing concious about the text.

Justin: This paragraph is still decent advice. Bearer tokens should not live forever. You can revoke based on various events as well, not just based on time. This should not have normative weight. Don't call it a MUST or SHOULD. This is contextual advice.

George: Half of what we do with 2.1 is to remove bad security advice. If we give the impression that bearer tokens do not expire then we need to add text. I am worried that this leads to bad practice.

Justin: If this remains a normative requirement, then I will not follow it. I have a number of larger-system use cases where the tokens take the role of API keys. Those keys get re-deployed when software gets updated. They are not rotated in the fashion of normal token changes. They are not re-issued using regular protocol tooling. In our security model it makes sense. That's the reason for my suggestion.

Aaron: If you are not using RFC 6749 tooling, then it is an API key.

Vittorio: In a world of continuous authentication, there are various techniques for revocation. If those circumstances do not arise then the token can (in principle) live forever. We need to do the work to create sensible criteria.

Aaron: You are saying that the AS must have a way to revoke a token, if it wants to do so.

Vittorio: It is difficult to find normative language that covers all cases.

Aaron: Don't hardcode the tokens but allow revocation.

Vittorio: We can enumerate a few ways to revoke the tokens but we cannot be exhaustive.

George: I am hearing the difficulty of coming up normative text. We have to make it more clear in 2.1, e.g. via examples of ways on how to do this. We hopefully do a better education (even if it is not normative).

Pieter: I don't have a problem with tokens that do not expire. You will have to define ways to get to the endstate though. You might need to give guidance on what "limited" means. Unlimited lifetime is a very long time. Guidance for developers on how to define policies would be good.

issue #106: requirement to support 3 redirect URI methods

This is a requirement from the native apps BCP (private URI schemes, claimed https URLs, loopback interface).
Some specifications do not permit the use of certain URI methods. Question: Should the request to support all three URI methods relaxed.

Daniel: Some deployments do not support native apps. if you supply native apps, then you do not need to any of these URI methods.

Vittorio: We have customers that do not want to use the loopback interface. Hence, I would not want to force people to implement certain URI methods.

Brian: It is not uncommon for profiles to further narrow what is allows. FAPI also does not allow client secrets. It would be better to qualify it.

Mike agrees with Brian.

Dave: Agree with Brian. The wording needs to change. It is high likely that ASs that do not support native apps need to support these URI methods. Unless there is a good reason, do no use private URI schemes.

Aaron: This is another option to make private URI schemes optional. I am not sure why all three are required by the AS. Maybe that was not the right goal to begin with.

George: I am in favor of ordering of best security practive starting with the claimed https URL.

Aaron: It sounds like we are going to re-order the list, claimed https URL first, loopback second, and private URI schemes. Remove the sentence for the mandatory-implementing all three. Maybe adding text of why you want to have the private URI scheme at all.

There are a lot more issues in the Github repo at github.com/aaronpk/oauth-v2-1/issues

credentialed client

Brian: There was a discussion about the credential client type but nothing has happened. Was it forgotten? Should I create an issue?

Aaon: I have forgotten about this issue. If you file the issue then we can make sure that it is tracked.

Here is the issue about “credentialed” client concept: https://mailarchive.ietf.org/arch/msg/oauth/emOUhFGpiOUrIjHfOssFv4HucEc/

Justin: Brian is not entirely wrong. The term is a bit confusing but Aaron is 100% correct that we have more than confidential client that are statically configured clients and then public clients. We need to acknowledge and enbrace this concept. I think we may want to approach this in a slightly different way such that the definition of the confidential client is a client that has a secret it can keep. A public client does not. Then, there is an another dimension where the confidential client became in possession of the secret. This might lead to a better desription by expanding the reach of what the confidential client means. If we want to refer to a sub-category, then we should come up with a different term (e.g.

native app vs. web server vs. MODERNA and open banking). In all places we say confidential client and in a separate section we explain the term.

Bikeshedding all the categories will be a mess but we have to go through this.

Aaon: I thik this is doable.

Brian: I think Justin proposed what I am proposing. The names currently are pretty bad but we are stuck with them. We cannot change them. We need to better explain the sub-categories.

Aaon: One other angle is whether the user gets prompted for consent. It is not just about authenticating the client and how the credential has been provisioned. It is also about how much the client is trusted by the AS.

Brian: A client registered via an open portal is not different from a cient registered via dynamic client registration.

Dave: I am agreeing with Brian and not all confidential client are the same either.

George agrees with the statement and provides examples.

David Waite provided feedback from remote, which was inaudiable.

Brian clarifies that David was saying that it is better to re-introduces completely new terms. Brian argues that the terms are used throughout various documents it is important to keep the names, essentially contradicting David's remark.

Vittorio is a bit concerned about re-defining the terms and to generalize the "confidential client" term to any client that is provisioned with a credential.

Brian: The generalization matches the reality: Confidential clients are clients provisioned with credentials.

Vittorio: I am not sure about the reality comparison. A Twitter client on Windows has the same client id on every version. Only later it gets an instance-unique credential provisioned.

George: I don't want public client refer to clients that mint a key pair, store it on a TPM, and then use it later. Public keys are those that cannot protect their secrets.

Vittorio: Twitter can use public clients and later when needed an individual instance can be identified.

Justin: Vittorio and I agree – in different directions. I would call the Twitter client as a confidential client that is just not statically configured with the credential. A selling point

for dynamic client registration was to take a public client and then to turn it into a confidential client. A confidential client is presenting a credential in some form. Whether you cange the client id or not, this is a separate discussion. If I decide to give everyone the same client id and then differentiate them based on keys (which is what Twitter does). This should not be wrapped around the client identifier.

Aaron: There are many aspects that are independent of whether the client has a secret or not. One is about simplifying the terms and then there are other aspects to talk about.

Justin: We need to make a distinction when there are differences. In the discussion here there are considerations about what happens on top of OAuth and not within OAuth.

George: Where do we have the larger discussion about “models”? This would be useful for developers to understand that there are multiple models and to discuss the tradeoffs.

Rifaat will send the info about the side meetings to the list again.

Thursday’s Agenda

Device Code Flow - Pieter (40 min)

Talks about social engineering attacks related to the device code flow. The problem is with the communication between the device and the smart phone/tablet where the user completes the transaction. User is tricked to enter the code into an attacker-controlled device. Versions of CIBA may also be vulnerability to this attack.

Filip also speaks about a social engineering attack he encountered.

Daniel shows a video illustrating the attack based on a user, who is willing to log into a website.

What can be done? (None, implementer guidance, new protocols)

Tim (Microsoft): To solve the use case we have to use WebAuthn and a new device flow. We have to start the discussion about getting the device code flow.

Justin: How does it work?

Tim: It uses proximity with Bluetooth.

Vittorio: WebAuthn is a stepping stone and we have an opportunity to be more descriptive. We could describe how Bluetooth is used to transfer context.

Tony: I guess you are talking about Google's authentication, it does not have much to do with WebAuthn. It is a FIDO solution. It has severe usability problems. It is probably not the right approach. There are both security and usability issues.

Justin: The composed URI is something I wrote for the device flow. There are very different interaction modality. There is something the user has to type and a URI being sent to the user. A first steps needs to be to separate the two. Sending a lot of information via Bluetooth is not the device code flow anymore. If you really have to use the device code flow, then we should only let you do a few selecting things.

Hans-Joerg: In Pieter's description he said that many developers do not know what they do. I would like to have a design patterns descriptions, something we have in programming languages.

Tim: In the next few months billions of phones will have the features available. From an IdP point of view it is just WebAuthn

Pieter: I hear that there is interest in this work. If someone is interested, reach out to me.

Interoperable Step-up Authentication - Vittorio (30 min)

https://datatracker.ietf.org/doc/html/draft-bertocci-oauth-step-up-authn-challenge

Use case: application asks for an initial set of claims. Then, later further rights are needed, which require further authentication steps by the user. For example, a request may exceed the current authentication level or the token needs to be fresher.

Problem is with the error situation that occurs.

Proposal is to create a new error code, require acr_values and max_age AS request parameters. Based on the acr_values the AS produces another token with the access token containing the acr claim.

Leif: Is acr_values multi-value?

Vittorio: Yes.

Justin: Why is the RS caring about the authentication levels? There is the great blog post, you explain that the AS decides about the authentication level. Why does the RS care about the authentication level?

Vittorio: We still care about the authentiation levels for the RS. It would be best if the RS translates this security requirement into a scope values but now we have the acr_values concept. Talking to the developers, they have asked for the feature "did the authentication use a smart phone"?

Justin: What about vector's of trust?

Vittorio: I leave this to Brian to answer.

George: For me this makes a lot of sense. If the RS and the AS are from the same domain, this could work fine. For cross-domain use cases, there is extra complexity there and we could decare that out of scope.

Vittorio: Great point. acr is "by reference" and both parties need to understand what it means. This might be enough of a scoping mechanism.

Brian: There is some agreement on the level of acr (e.g. pair-wise agreement and ecosystem agreement). There is not necessarily a limit to a single domain. I disagree with this characterization.

Filip: The client and the RS need some level of trust. I don't know whether there is anything in it but might be worthwhile to think about it.

Vittorio: The semantic of acr_values and max_age is very well define. We need to make sure that we do not enable all extension. We can define the semantics of these parameters narrowly enough

Leif: There is an IANA registry for access authentication context (establised by RFC 6711); it has all the globally recognized context classes. This could be used. There are certain things that are globally recognized. We should update that RFC and to apply it to OAuth and OpenID Connect.

Dave Robin: The permissions seem to be no longer scope based but rather a combination of scope and acr-values. Why not use scopes? What do the additional scopes do not give you.

Vittorio: Scopes have enough expressive power to do all this. We have acr_values already defined and we can use them. We believe this would be clearer. Developers wanted to avoid the cases by step-up-authentication by requesting as many scopes as possible.

Dave: This is similar to Resource Indicator.

Brian: Modelling the Resource Indicator with scopes is hard. We allow a more explicit indication.

Mike: Re-inforcing what Leif said. Also agree with Brian.

Vittorio: If you think the problem is worth solving, a draft is available.

Rifaat: I encourage you to update the document and to submit a new version.

Filip, Pieter, and Robin volunteer to review the draft.

Libraries - Daniel (40 min)

https://mailarchive.ietf.org/arch/msg/oauth/h9_Ki1UYT8sS0xKqGrzWI6yHaNA/

Daniel believes that there is a lack of “good, modern and universal” OAuth client libraries. Currently there are very few of those libraries and customers have to create custom implementations.
Many unmaintained implementations, incomplete implementations, and often undocumented (one has to look at the code to understand whether the latest security implementations are followed).

Due to the lack of the API, vendors have to provide extensive documentation and developers just write them based on the documentation.

In practice, you have to find out how to configure a library (such as authorization endpoint, token endpoint URL, userinfo endpoint, supported grant type, method of client authentication, what security mechanisms to use and how, etc.). Without server meta-data this is pretty tedious.

Consequence: unnecessary fragmentation, slow adaption of new specs, developer frustration, time & money.

Vittorio: I have owned SDKs for several years. The conformance tests is a good idea. Talking about OAuth library is a bit misleading since they are there to help accomplish some goal, which is often disconnected from the protocol. We, for example, don’t talk about sign-up but the libraries do. SDKs often don’t expose the low level. It may be hard to configure aspect. There are also various jobs, such as sign in. There are differences for libraries depending on how various talks are done with specific vendors. The meta-data topic: We are happy that Apple supports OAuth but they are not supporting meta-data. You didn’t suggest that we write our own libraries.

Daniel: I don’t think we should develop libraries. Blessing is something else. There is not always one use case but I believe a library can be created to dig deeper. Compare with HTTP client libraries to make GET requests, to add a cookie or to add specific headers. There should be a way to write libraries to allow the specific requests to be made. Users should not be bothered with creating HTTP requests themselves.

Justin: Want to echo Vittorio. The use of an OAuth library is always contextual. HTTP libraries are not. OAuth is a not a complete layer separation. I also wish the world would be better. There are, however, problems. OAuth is not a single protocol. You have to implement several different libraries. I have maintained a Java library for years. The use of it will be drastically different depending on the flow. Developers want to get the end state. When we put our book together to ensure that our examples were not provider specific.

The first review was: this is useless because it does not help to connect to Github. The audience for the general purpose library suite is actually SDK developer.

Mike: In the OpenID Connect world, we can host a list of certified implementation. This means that we have a quality bar. As Daniel points out, there is no quality bar. We could, however, list quality implementation. Aaron P. has oauth.net/code and he could list implementations.

Hans Joerg: I appreciate the discussion. I think this is a generic issue applicable also to other working group. OAuth libraries may be included in SDK. For less-known provider, you will have to use a generic library. There are also security problems. Maintaining OAuth credentials get more challenging with different libraries. It would be nice if SDK developers to offer a token from a third party library.

Pieter: We have looked in implementations and there are implementations made again and again. We should define standard test cases and I am very supported of this idea.

Joseph (Authlete / OpenID Foundation): There are some strange stuff in libraries out there. Even in open banking the security features/checks are not good. The profiles and the meta-data would make things a lot easier. Getting a library though a certification program, like OpenID Connect, is a non-zero effort. There are a lot of OAuth libraries that are very old. There has to be a collaborative effort among the authorization server vendor. Also the documentation has to say consistent things.

Dave Tonge: Agree with Joseph and disagree with Justin about the context difference. You don't implement an HTTP library and you do not implement a JSON library either. I don't understand why one would implement an OAuth.

Daniel: There is maybe an 80% use case here, which we could focus on.

Aaron: Happy to put more work on oauth.net website to better show the websites. I have not done a lot of maintainance in terms of auditing libraries. There are a lot of options available. A conformance test would certainly help. Happy to take suggestions.

Justin: Appreciate that Dave disagree with me and then made the same points I made. The target is not the same as general libraries. You are building this to be included in something else.

Daniel: I will follow-up on the mailing list on this topic.

PKCE in the Security BCP - Daniel (10 min)

Daniel reported from a side-meeting on the PKCE. PKCE is in the OAuth security BCP. It is a solution or protection against authorization code misuse (stealing authorization code and swapping an authorization code).

Daniel shows a table with different configurations and for public clients PKCE is the only way to mitigate certain attacks.

Hence, the wording in the securiy BCP document will be kept and further text will be explained to motivate the text.

Mike: Thank you for putting this together. I have thought about some of the OpenID Connect flows and without the PKCE there is no mitigation. I believe there are still a few details to be worked out. Mike voluntered to review the text.

Aaron: There are two attacks to talk about: code injecting and stealing a token. That's why I created the table.