SAML2int v2.0

GitHub source: https://github.com/KantaraInitiative/SAMLprofiles/tree/master/edit/saml2int

Rendered version: https://kantarainitiative.github.io/SAMLprofiles/saml2int.html

Issue tracking table

	Reporter	Issue	Submitter Comments	Response(s)	Disposition
1	Rainer Hoerbe	NA	The first paragraph in the introduction should contrast the deployment profile with an implementation profile, and reference the SAML Implementation Profile for Federation Interop for this purpose. The difference between both types of profiles is not widely understood.	Sounds sensible to the group. Slot in after first paragraph of introduction. Nick volunteers to propose language. Status update 2019-06-11, addressed in commit https://github.com/KantaraInitiative/SAMLprofiles/commit/376ce65dccfd838bd5676712682602f14ca4a588	Accepted
2	Rainer Hoerbe	SDP-MD02	I do not understand the explanation for [SDP-MD02]. If PKI with path validation is being used, there would be no hindrance to roll out new keys, even if metadata and assertions use the same key. I have seen a IDPs that publish their own metadata and the well-know location using the same signing key as for assertions.	(Scott) I think you may be correct about that and that the text is written with a presumption of the verification approach, and if we didn't specify that (and I don't think we did), it's open to methods that wouldn't have the problem we were concerned about. I think it needs work. Good catch. In a closed environment where you have control of the trust anchors, this would work. You could obtain metadata signing keys from a federation and publish signed metadata locally. This is correct in theory but not in practice - PKI doesn't federate beyond a closed ecosystem. We are trying to leave too much open, need to say how you trust the signature. Need to give a couple of examples, in this example the key would have to be different, in this one, the key would be the same. It's the binding of the key to the entity that's the problem with the model Rainer is talking about. The qualifier in the italicized text in MD02 is what we need to pull up into a positive requirement.	Accepted
3	Rainer Hoerbe	SDP-SP03	"This will typically imply that requests do _not_ involve a full-frame redirect ..". In my understanding it is the other way round; in Javascript terms one has to execute "document.location = url;" Also, what is the approach for single page applications?	(Scott) Ouch. Yeah, that's backwards. (re: SPA): Generally AJAX use has to be governed by more intelligent server side signaling and code able to detect a loss of session without being inadvertently thrown into a SSO loop, and that's not even just due to framing but simply the lack of a UI to handle the redirect when it happens at the wrong time. We'll fix the backwards part.	Accepted
4	Rainer Hoerbe	SDP-SP23	I think that the division of IDP-discovery into disco-UI and preference persistence is a significant improvement over the current IDP-Discovery spec, fixing the issue that embedded discovery results are not shared across SPs. See the RA21-proposal: https://groups.niso.org/apps/group_public/download.php/21376/NISO_RP-27-2019_RA21_Identity_Discovery_and_Persistence-public_comment.pdf. Rumor has it that Leif implemented it in pyFF.	(Scott) The discovery spec that's referencing never addressed UI or persistence, it's an interop protocol only, to enable a discovery solution to be injected into the flow, whatever solution it might be. We should ask Rainer to clarify.	The group believe that there is no strong consensus on best practice for this aspect of discovery.
5	François Kooman	SDP-ALG01	I'm trying to understand the RSA-OAEP encryption requirements for IdPs / SPs. > *The following default digest algorithm MUST be used in conjunction with the above key transport algorithms (the default mask generation function, MGF1 with SHA1, MUST be used):* > > *http://www.w3.org/2001/04/xmlenc#sha256 [XMLEnc]* It seems most IdPs use SHA1 for both the MFG1 and digest? So, this profile requires you to use SHA1 for the MFG1 and SHA256 for the digest. Any reason why it is not SHA256 for both? Also, why not require MGF1 with SHA256: `http://www.w3.org/2009/xmlenc11#mgf1sha256` as algorithm identifier? Now it is not clear that SHA256 was used for the digest? Probably I am missing something here... (Github Issue #129)	(Judith) I read the parenthesized reference to the default mask generation function to be a reiteration of a requirement stated elsewhere, particularly XMLEnc's §5.4.2 statement that "As described in the EME-OAEP-ENCODE function RFC 2437 [PKCS1, section 9.1.1.1], ... using the mask generator function MGF1 (with SHA1) specified in RFC 2437." If i am correct, i wonder if rewording as follows would be more clear > *Key Transport (the default mask generation function, MGF1 with SHA1, MUST be used)* > > *http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p [XMLEnc]* > > *http://www.w3.org/2009/xmlenc11#rsa-oaep [XMLEnc]* > > *The following default digest algorithm MUST be used in conjunction with the above key transport algorithms :* > > *http://www.w3.org/2001/04/xmlenc#sha256 [XMLEnc]* (Scott) There is definitely clarification needed, it reads very badly now...but most IdPs have long since stopped using SHA-1 for general usage, the MGF1 case is an exception and was left as is for interoperability. It's not that unusual for libraries to lack support for any MGF pluggability. If there are security implications for use of SHA-1 there, I'm not aware of them.	Accepted
6	via Rainer Hoerbe	SDP-IDP07		Eric Goodman wrote on 6/6/19 12:22:	Accepted

I received a comment from an Austrian government agency wrt to the required authentication challenge of Forced Re-Authentication. They are using other mechanisms than passwords, such as Kerberos and client certificates. They write: "In such use cases the concrete meaning of this feature is unclear. Beside the fact, that authentication does not involve user interaction in every case, using re-authentication for an improved “Are you sure?" Dialog results in bad user experience. The logon screen of an IdP does not explain what is going on. Other protocols should be used for this use case. For example with the current Austrian governmental E-ID solution it is possible to sign a text or an XML-Document. Only protocols like that are providing an improved non-repudiation, by binding the information the user has to acknowledge with a signature.“ I think that one could argue, that 'previous session' on a managed device with a screen lock is a good-enough proof of presence.

> *Other protocols should be used for this use case. For example with the current Austrian governmental E-ID solution it is possible to sign a text or an XML-Document.*

The saml2int standard can't make recommendations around potential non-saml solutions. So I think this argument is orthogonal to the requirement in the profile. Of course ForceAuthn is not going to be for many specific authentication purposes as locally developed "fit for purpose" solution, especially in communities that can dictate SP's implement to that alternate specification. That just doesn't seem like a strong argument that saml2int should NOT define some baseline, SAML-based criteria be supported for the cases where this is not the case. So I think the argument for removing it from the profile needs to be based on "there is little or no value to the feature in SAML (or in the SAML profile) overall", and not "I can design a different protocol that is a better match for my needs". --- Eric

Cantor, Scott wrote on 6/5/19 17:33:

On 6/5/19, 5:18 PM, "WG-FI on behalf of Rainer Hoerbe" <wg-fi-bounces@kantarainitiative.org on behalf of rainer@hoerbe.at> wrote:

> "In such use cases the concrete meaning of this feature is unclear.

I wouldn't agree with that at all, but it's not that important for the purposes of the issue.

> Beside the fact, that authenticati on does not involve user interaction in every case, using re-authenticati on for an improved "Are you sure?" Dialog results in bad user experience.

ForceAuthn is often a bad user experience, that is certainly true.

> The logon screen of an IdP does not explain what is going on.

I don't think anybody can argue that every IdP in the world "does not" do this, and certainly many *could* do it. Maybe there should be guidance saying one should.

> I think that one could argue, that 'previous session' on a managed device with a screen lock is a good-enough proof of presence.

I probably agree, because that's part of the deployment. Maybe the solution is to supplement the text that's there to explain the broader scope. It's not a necessity that IdP *software* know anything about what's happening to be configured to make the right things happen. -- Scott