Consultation: Error Handling

This consultation closed on 14th May 2020 at 17:00 CEST. The REFEDS Steering Committee held an e-vote and approved this specification on 1 July 2020.

Background

The goal of this specification, "SAML V2.0 Metadata Deployment Profile for errorURL Version 1.0" is to offer specific guidance on how a service should inform users of non-technical shortcomings of logins.

As indicated in the specification:

> *This profile provides a set of conventions around the use of the attribute that extends the usefulness of the feature such that Service Providers can make more effective use of the feature when encountering conditions that can plausibly be remedied by the user's Identity Provider. The profile is compatible with existing uses of the attribute such that Service Providers can easily determine if the profile is supported by the Identity Provider.*

The document contains four main sections:

- An introduction that describes the purpose of the specification.
- A profile description that defines a convention for the syntax for the errorURL
- User interface guidance
- Examples

The specification has been prepared by the REFEDS Best Practice around Error Handling Working Group and is now being made available for public consultation according to the REFEDS Participants Agreement.

Overview

Participants are invited to:

- Review and comment on the proposed specification for SAML error handling.

Following the consultation all comments will be taken back to the REFEDS Error Handling working group for review and if appropriate the White Paper will then be forwarded to the REFEDS Steering Committee for sign-off and publication on the REFEDS website as per the REFEDS participants agreement.

The document for the consultation is available as a Google doc in Suggestion mode or as a pdf attachment. All comments should be made on: consultations@lists.refeds.org, added to the google doc as a suggestion or added to the change log below. Comments posted to other lists will not be included in the consultation review.

Change Log

We recommend adding changes to the Google doc directly via suggestion mode. For those who would prefer not to use Google, please use the change log below referencing the pdf attachment.

	Line / Reference	Proposed Change or Query	Proposer	Action / Decision (please leave blank)

1	**2.4. returnURL**	What seems to be missing from this specification is a way to let the user continue to do what he/she came to do: use the SP. If the federate login via the institution did not work, e.g. because of MISSING_ATTRIBUTES, the user may now end up at an Idp error page. Regardless of the root cause of the error, it is unlikely the IdP will be able to mitigate this problem within the time the user wants to get access to the SP. Effectively we have now silo-ed the user into a place where there is no obvious way to continue, and the user is 'lost' to the SP, as the SP is several redirects away. That is a rather poor user experience. However, the SP might be offering other means of login to the service, e.g. social or guest IdP login, or perhaps access to the SP with less privileges. With this proposal, there is now no easy way for the user to return to the SP and use an alternative. I would therefor propose to add a capability to let the SP add a 'return URL' to the Optional Placeholders so a user may, after having been informed by the IdP of the issue, may return to the SP and continue to work. In this way the IdP error page could present a link or button to allow the user to "Return to the service'. === **2.4. returnURL** An optional URL set by the Service that the IdP can present to the end user to allow the user to continue using the Service via some other authentication method. The returnURL MUST appear within the query string of the URL. This requirement allows the URL-encoding rules to be less ambiguous. === It is already very inconvenient the user was not able to use federated login. We should not punish the user double by blocking him from continueing to do his work.	Niels	We send the user back to the IdP because, with the user already blocked, the SP cannot do anything further - there is no reasonable way to continue from the SP's perspective. We have added the following to the User Interface Guidelines to support a need for the SP to maintain control of the user agent: If the SP has other options to continue despite the detected login problem (e.g. continue as is with less attributes/authorization or use some other way to handle authentication), the SP MAY give options to the user in addition to the er rorURL link. The SP MAY cause the errorURL to be opened in a new window to maintain control of the user agent. However, the SP MUST NOT use frames to present the errorURL.
2	2.3.4	> *If ERRORURL_CODE is "MISSING_ATTRIBUTES", this value if present SHOULD be set to a space-delimited list of the names of the missing attributes and, if appropriate, URIs of the applicable entity categories.* The "name of missing attributes" might lead to some SPs mentioning the friendly names or their local human readable attribute names. This might lead to inconsistencies. It would be more consistent if the specification said that this value should present a space-delimited list of attribute names in urn:oid:#LDAP OID# format or as declared by the SP in its RequestedAttribute elements.	Lukas	The working group considers the lack of precision regarding the friendly name to be a positive in that it helps make the specification protocol agnostic. The friendly name is under the control of the IdP; if the SP passes that back then the IdP will be well positioned to handle this. If the SP creates a new name, that might create more problems than it solves The alternative would be to put in more language specific to every protocol. We could be more specific with attribute names to include the protocol, and direct away from friendly names. It will be longer names that humans can't read because they will be OIDs. No change will be made to the specification.
3	2.2.1.1 and 2.2.1.3	It looks like there is some overlap between the code definitions MISSING_ATTRIBUTES and AUTHORIZATION_FAILURE. The definition for MISSING_ATTRIBUTE does not require that it's only attributes for authorization, so it could be thrown when attributes essential to the operation of the site are missing (no givenName or email address). The definition of AUTHORIZATION_FAILURE also says that it can be thrown when there's a missing attribute or value. So if there is a missing attribute that is required to allow the SP to make an authorization decision, it could be flagged with either code. Is this a problem? It could be resolved by clarifying - An error condition for MISSING_ATTRIBUTES is determined before looking at any of the values, and that it relates to any attributes, not just those for authorization. - stating that AUTHORIZATION_FAILURE is when the SP has received all the attributes it needs, but the values do not allow authorization.	Alex	The working group agrees that the name "MISSING_ATTRIBUTE" is unclear. We have changed "MISSING_ATTRIBUTE" to be "IDENTIFICATION_FAILURE" and clarified the description of the code.
4	2.2.1.1, 2.2.1.3	It's not clear to me when an SP should send MISSING_ATTRIBUTES or AUTHORIZATION_FAILURE in case no attribute (or attribute value) that would allow authorization to pass has been found. 2.2.1.1: The SP did not receive one or more attributes or values it requires 2.2.1.3: The user is not authorized to access the SP. This may be caused by [...] entitlements, affiliation or missing attribute or value Ignoring the grammatical issue of the sentence in 2.2.1.3, does that man that an SP should only ever send MISSING_ATTRIBUTES if the missing attributes are NOT being used for authorization (but strictly required for other reasons)? E.g. the application needs a name in order to create a local account but that's missing. If so then I think this would need more text to spell that out. **"This [authz failure] may be caused by an inadequate assurance level (when expressed independently of authentication), entitlements, affiliation or missing attribute or value" the relevant part to my question being (my upper-casing) OR MISSING ATTRIBUTE OR VALUE at the end.	Peter	The working group agrees that the name "MISSING_ATTRIBUTE" is unclear. We have changed "MISSING_ATTRIBUTE" to be "IDENTIFICATION_FAILURE" and clarified the description of the code.

5	2.3.4	If sending "names of the missing attributes" does that imply a certain NameFormat? E.g. are friendlyNames possible even if e.g. URI naming is expected on the wire? Does the spec make a recommendation here? And what attribute names would I send if the SP requires only/at least one of several possible attributes, which is the common case due to multiple identifier types in use? Would the SP send all of the missing attributes that are would be acceptable? Only one (and which one, e.g. CoCo says to use the least privacy invasive)? That's the same issue as with SAML 2.0 Metadata isRequired, of course. Also, what does the part about sending the "URIs of applicable entity categories" mean (lines 111 ff.). "If ERRORURL_CODE is “MISSING_ATTRIBUTES”, this value if present SHOULD be set to a space-delimited list of the names of the missing attributes and, if appropriate, URIs of the applicable entity categories" In what cases should the SP be sending what entity category URIs to the IDP? In case the SP already lists some categories in its metadata and the IDP... • claims to support some category but didn't? (That's of course an error by the IDP that would ultimately lead to the IDP losing the support category.) • didn't claim to support (some/any) categories and also in fact didn't? Not sure how useful it is to tell the IDP what categories an SP wants it to support when the SP already has all that info in its metadata? (Of course the same could be said about missing attributes but there are many more reasons that didn't happen, I think.)	Peter	We believe the first half of this comment has been resolved with the changes made around “MISSING_ATTRIBUTE” (now “IDENTIFICATION_FAILURE”). Regarding the utility of telling the IdP what entity category the SP wants when it' s declared is that this allows the IdP to tailor a page to respond to this particular issue. For example, if the IdP is explicitly not going to give an SP the attributes because of institution-level restrictions (e.g., campus registrar restrictions, FERPA prohibitions) they can tailor their error page for the user to explain the situation to the user.
6	4 (Examples)	I think the distinction between the literal string listed in the IDP'S metadata ("IdP errorURL") and the dynamically generated URL value that's being used by the SP ("Processed errorURL") could be made more clear. If I did not understand/paraphrase their meaning correctly above then doubly so. ;) Maybe also show an example with a (permitted, AFAIU) mismatch between the parameters the IDP lists and the ones the SP sends back, e.g. by sending the optional, unused "variables" /"template strings" to the IDP verbatim. (That's more likely than the SP parsing the errorURL and stripping off any parameters it's not willing or able to include itself.)	Peter	The example for “IDENTIFICATION_FAILURE” has been extended.
7	3	*> > As long as it does not do that I'll assume that the whole errorURL* *> > spec only serves for the IDP to log/store that in some backend system* *> > for potential later analysis (in the few cases where the SP "MAY"* *> > chose to even render that link, *and* the subject also choses to* *> > follow that link).* *>* *> That's exactly the intent. Anything else is never going to work* *> widely because there's too little actionable information for a user* *> to act on here. It's an attempt to make errorURL at least a little* *> more potentially useful to some IdPs if they know enough about their* *> operations to recognize the likely reason why certain problems* *> occur.* OK. I'd suggest to change section 3 then to allow for the SP to directly send that report to the IDP, without needing the subject to click on a link.[1] At least then the IDP would get reports to its published errorURL for all issues "reportable in principle" by the SP, not only those were the subject found, understood and/or could otherwise be bothered to click on a link to the IDP's errorURL.	Peter	The intent of the specification is to support front channel communications only. If the errors are automated in a backchannel fashion directly between the IdP and SP, every time the user shows up without the necessary attributes, the reports back will never stop. Doing this the direct-to-IdP way is spamming. The reverse is also true. The user having to click on something is a natural throttle to keep this to a bare minimum. No change will be made to the specification
8		Wouldn't hurt to spell that out, that MISSING_ATTRIBUTES should only ever be reported for cases that do not involve authz (or in a positive statement, if one can come up with one)? I.e., whenever authz failes due to missing attributes the AUTHORIZATION_FAILURE code takes precedence over the MISSING_ATTRIBUTES one.	Peter	We believe this has been addressed with the changes to MISSING_ATTRIBUTES (now replaced with “IDENTIFICATION_FAILURE”). An additional example will be added to section 4.