Publik - Support #33232

équivalent "SessionNotOnOrAfter" en oidc ?

19 mai 2019 08:43 - Frédéric Péters

Statut:	Nouveau	**Début:**	19 mai 2019
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Club:**	Non
Planning:	Non		

Description

En SAML on cale l'expiration d'une session d'un SP sur ce qu'authentic fournit dans <SessionNotOnOrAfter>, des pistes pour obtenir ce comportement avec oidc ?

Demandes liées:

Lié à Authentic 2 - Development #33241: calculer l'attribut exp de l'id_token...	**Nouveau**	**19 mai 2019**
Lié à Authentic 2 - Development #33240: poser un cookie browser state	**Nouveau**	**19 mai 2019**
Lié à Authentic 2 - Support #33242: implémenter la section « 4. Session Stat...	**Nouveau**	**19 mai 2019**
Lié à Authentic 2 - Support #33243: implémenter un équivalent SAML à la vue "...	**Nouveau**	**19 mai 2019**

Historique

#1 - 19 mai 2019 09:20 - Paul Marillonnet

Outre les techniques à base d'iframe pour l'échange d'informations relatives à l'état de la session entre le fournisseur OIDC et le RP, le brouillon de doc OIDC sur la gestion des sessions précise que "_An ID Token typically comes with an expiration date. The RP MAY rely on it to expire the RP session_[1]".
Ce que tu décris dans le ticket semble pouvoir être géré à l'aide du champ de timestamp d'expiration sur l'ID token, mais, contrairement aux iframes, ne couvrira pas le cas où l'usager termine prématurément la session côté fournisseur OIDC.

[1] Sections 4 ; 4.1 et 4.2 de https://openid.net/specs/openid-connect-session-1_0.html#ChangeNotification

#2 - 19 mai 2019 10:18 - Benjamin Dauvergne

- Projet changé de Authentic 2 à Publik

- Club mis à Non

De mon coté je serai plutôt pour implémenter la gestion des sessions via iframe, et propager cette technique à SAML, c'est beaucoup plus sécurisant.

Je déplace ce ticket sur Publik parce qu'il faudra plusieurs tickets d'implémentation :

- un pour gérer finement l'attribut exp des id_token, pour l'instant c'est géré par ce code dans src/authentic2_idp_oidc/views.py :

```
def idtoken_duration(client):
    if client.idtoken_duration:
        return client.idtoken_duration
    return datetime.timedelta(seconds=app_settings.IDTOKEN_DURATION)
...
    start = now()
...
            'exp': timestamp_from_datetime(start + idtoken_duration(client)),
```

- une autre pour proposer l'iframe de gestion des sessions OIDC
- un dernier pour inventer une implémentation équivalente avec OIDC

#3 - 19 mai 2019 19:18 - Benjamin Dauvergne

- Lié à Development #33241: calculer l'attribut exp de l'id_token comme le sessionNotOnOrAfter de SAML ajouté

#4 - 19 mai 2019 19:18 - Benjamin Dauvergne

- Lié à Development #33240: poser un cookie browser state ajouté

#5 - 19 mai 2019 19:20 - Benjamin Dauvergne

- Lié à Support #33242: implémenter la section « 4. Session Status Change Notification » de la spécification OpenIDConnect Session 1.0 ajouté

#6 - 19 mai 2019 19:25 - Benjamin Dauvergne

- Lié à Support #33243: implémenter un équivalent SAML à la vue "check_session_iframe" d'OIDC ajouté