Authentic 2 - Development #82729

IdP OIDC : rendre factultative la case à cocher « Ne plus redemander » sur la page de consentement

23 octobre 2023 15:15 - Thomas Noël

Statut:	Fermé	**Début:**	23 octobre 2023
Priorité:	Normal	**Echéance:**	
Assigné à:	Thomas Noël	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

Dans la fenêtre de consentement vers un service OIDC, on a une case à cocher « Ne plus redemander ».

Elle est en dur dans src/authentic2_idp_oidc/templates/authentic2_idp_oidc/authorization.html :

```
<p class="a2-oidc-authorization-form--do-not-ask-again">
  <label for="id_do_not_ask_again"><input id="id_do_not_ask_again" type="checkbox" name="do_not_as
k_again" value="on"/><span>{% trans "Do not ask again" %}</span></label>
</p>
```

Si la case n'est pas cochée, aucune autorisation OIDCAuthorization n'est enregistrée, c'est l'effet voulu, bien sûr. Mais cela pose des problèmes fonctionnels lorsque, plus tard, on veut que Publik contacte le "sub" renvoyé au service (par exemple pour avoir "les demandes en cours sur tel autre logiciel" relié par OIDC).

Je proposerai donc que sur un client OIDC on puisse avoir des paramètres de configuration supplémentaire :

- lors du consentement, toujours enregistrer une autorisation longue (ne plus proposer la case à cocher « ne plus redemander »)
- la durée d'autorisation longue est le xxx jours (actuellement 365 en dur dans le code) (ça sera celle si la case « ne plus redemander » est cochée, ou si le consentement implique autorisation longue)

L'idée sous-jacente est que le système /api/users/<user_uuid>/service/<service_slug> renvoie bien toujours le sub de l'utilisateur dès qu'il a accepté la connexion via OIDC (cf #79230)

Demandes liées:

Lié à Authentic 2 - Development #83013: IdP OIDC : remonter une erreur lorsqu...	**Fermé**	**02 novembre 2023**
Lié à Authentic 2 - Development #83024: IdP OIDC : ne pas accepter de valeurs...	**Rejeté**	**02 novembre 2023**

Révisions associées

Révision c491d6f8 - 31 octobre 2023 16:56 - Thomas Noël

idp_oidc: make "do not ask again" choice optional (#82729)

Add a « always save authorization » parameter in OIDC client configuration.
Also, allow a custom authorization duration.

Révision e0d257f6 - 31 octobre 2023 17:15 - Paul Marillonnet

translation update (#82729)

Historique

#2 - 24 octobre 2023 11:41 - Robot Gitea

- Statut changé de Nouveau à En cours

- Assigné à mis à Thomas Noël

Thomas NOËL (tnoel) a ouvert une pull request sur Gitea concernant cette demande :

- URL : https://git.entrouvert.org/entrouvert/authentic/pulls/154
- Titre : WIP: idp_oidc: make "do not ask again" choix optionnal (#82729)
- Modifications : https://git.entrouvert.org/entrouvert/authentic/pulls/154/files

#3 - 24 octobre 2023 14:19 - Robot Gitea

- Statut changé de En cours à Solution proposée

#4 - 30 octobre 2023 16:31 - Paul Marillonnet

Alors, si on résume un peu l'affaire :
· il y a eu #79230 où, pour faire les choses dans le respect du consentement de l'usager, on ne fournissait un sub que si l'usager avait consenti à transmettre ses données au service identifié :

```
        if not auth_manager.filter(user=user, expired__gte=now()).exists():
            return {}
        return {'id': utils.make_sub(self, user)}
```

· maintenant, on souhaiterait pouvoir connaître ce sub quand bien même aucune autorisation existerait en base.

J'ai l'impression que, plutôt que de faire du billard à trois bandes (en se disant qu'on va masquer la case à cocher pour stocker toujours cette autorisation et donc avoir le sub à coup sûr), on pourrait, dans la réponse retournée par l'endpoint de #79230, ajouter un champ pour définir s'il existe une autorisation connue ou non.

Et donc à l'appel de /api/users/<uuuid>/service/<slug> on aurait une réponse du genre :

```
{
  "err":0,
  "result":1,
  "data": {
    "service": {
      "slug":"slug-du-service",
      "name":"Nom du service",
      "type":"LibertyProvider" (SAML) ou "OIDCClient" (OIDC)
    },
    "user": {
      "id": "le sub ou le nameid",
# <- possiblement pré-généré, notamment si l'usager n'a pas encore d'autorisation explicite
      "existing_authz": true/false
# <- en fonction de si auth_manager.filter(user=user, expired__gte=now()).exists()
    }
  }
}
```

Cela me paraît un moindre mal plutôt que de désactiver la possibilité pour l'usager de donner une autorisation ponctuelle pour un seul SSO, mais peut-être je loupe un truc ?

#5 - 30 octobre 2023 17:10 - Thomas Noël

> maintenant, on souhaiterait pouvoir connaître ce sub quand bien même aucune autorisation existerait en base.

Pas tout à fait : je veux connaître le sub si une autorisation a déjà eu lieu.

Dans ma proposition, pour que ça soit très explicite, j'en fais une configuration, qu'on dise explicitement à Authentic d'enregistrer les autorisations pendant x temps (et pour les fous, genre, 10 ans).

Dans la réalité des usages, ça ne me semble pas être un gros sacrifice à la vie privée (mais je veux bien en débattre). Et ça reste parfaitement optionnel.

> (...) ajouter un champ existing_authz pour définir s'il existe une autorisation connue ou non.
> Cela me paraît un moindre mal plutôt que de désactiver la possibilité pour l'usager de donner une autorisation ponctuelle pour un seul SSO, mais peut-être je loupe un truc ?

Si on fait ça, alors le filtre user|get_user_by_service:"oidcclient" dans wcs/combo devra toujours renvoyer le sub, que existing_authz soit vrai ou pas (parce que sinon mon cas d'usage tombe à l'eau, où je veux obtenir le sub, je dois l'avoir, y compris des gens qui n'ont pas accepté l'enregistrement de l'autorisation)...

Et donc on va avoir ce sub y compris quand celui-ci aura été expiré (ou parce que l'utilisateur aura demandé une dé-fédération, ou autre), et ça m'ennuie un peu plus, parce qu'on ne pourra jamais couper le lien en flinguant les autorisations (ok, c'est assez illusoire). Mais surtout, ça ne permettra pas de savoir si la personne a déjà donné un jour son autorisation -- et ça c'est assez bloquant pour l'usage recherché.

Voilà pourquoi je pensais à ce patch un peu plus profond dans la gestion des autorisations.

Last but not least : pour les poweruser, si on garde "la possibilité pour l'usager de donner une autorisation ponctuelle pour un seul SSO" mais qu'on

lui montre par ailleurs sur Publik qu'on a gardé le lien entre Publik et le service tiers (le sub), alors je trouve ça un peu... crasse.

Note que pour aller dans le sens de ta proposition, je pourrais imaginer un filtre « user|get_user_by_service_toujours_toujours:"oidcclient" » qui se ficherait du existing_authz mais (a) j'ai pas d'idée de nom (b) c'est pas simple à expliquer (c) reste le pépin de l'usager qui n'a encore jamais lancé de SSO vers ce service.

#6 - 31 octobre 2023 08:20 - Paul Marillonnet

Ok ok je comprends mieux, et en particulier l'usage du filtre qui fait qu'il est plus important, dans ce cas ci, d'avoir un sub valide plutôt que de laisser à l'usager la possibilité de ne pas enregistrer son autorisation.

Je vais relire le patch proposé ici et qui part dans cette direction.

#7 - 31 octobre 2023 15:32 - Robot Gitea

- Statut changé de Solution proposée à Solution validée

Paul Marillonnet (pmarillonnet) a approuvé une pull request sur Gitea concernant cette demande :

- URL : https://git.entrouvert.org/entrouvert/authentic/pulls/154

#8 - 31 octobre 2023 16:56 - Robot Gitea

- Statut changé de Solution validée à Résolu (à déployer)

Thomas NOËL (tnoel) a mergé une pull request sur Gitea concernant cette demande :

- URL : https://git.entrouvert.org/entrouvert/authentic/pulls/154
- Titre : idp_oidc: make "do not ask again" choix optionnal (#82729)
- Modifications : https://git.entrouvert.org/entrouvert/authentic/pulls/154/files

#9 - 02 novembre 2023 08:43 - Paul Marillonnet

- Lié à Development #83013: IdP OIDC : remonter une erreur lorsque la sauvegarde systématique d'autorisation est configurée avec le mode sans autorisation ajouté

#10 - 02 novembre 2023 10:02 - Paul Marillonnet

- Lié à Development #83024: IdP OIDC : ne pas accepter de valeurs ridiculement élevées pour le délai par défaut de stockage des autorisations ajouté

#11 - 02 novembre 2023 18:14 - Transition automatique

- Statut changé de Résolu (à déployer) à Solution déployée

#12 - 07 janvier 2024 04:42 - Transition automatique

Automatic expiration