Authentic 2 - Development #52679

ldap: ne pas logger en erreur les soucis LDAP, seulement les rapporter en front

02 avril 2021 15:11 - Benjamin Dauvergne

Statut:	Nouveau	**Début:**	02 avril 2021
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

https://sentry.entrouvert.org/entrouvert/publik/issues/27146/

Ça ne sert à rien de recevoir des traces pour des TIMEOUT ldap, ce n'est pas notre faute, il suffit de le rapporter en front et de laisser les utilisateurs concernés signaler à leur DSI que c'est instable.

Historique

#1 - 02 avril 2021 15:27 - Thomas Noël

Petite remarque : c'est sans doute un peu délicat car on peut avoir un timeout sur le LDAP des agents alors que la personne qui cherche à se logguer n'a pas de compte LDAP (un usager lambda). Il faudrait afficher l'erreur LDAP seulement si on est sûr que le compte est dans LDAP (mais comme le LDAP ne répond pas...)

#2 - 02 avril 2021 15:37 - Benjamin Dauvergne

Comme on garde username/email a priori on sait, en fait je me disais aussi que le provisionning LDAP à la volée était une mauvaise idée de toute façon, on ne perd rien ou pas grand chose à ne créer les comptes que dans les synchros.

#3 - 03 avril 2021 01:10 - Thomas Noël

Benjamin Dauvergne a écrit :

> Comme on garde username/email a priori on sait, en fait je me disais aussi que le provisionning LDAP à la volée était une mauvaise idée de toute façon
> on ne perd rien ou pas grand chose à ne créer les comptes que dans les synchros.

Je ne sais pas si on a un flag "ne pas faire de provisionning" dans la config LDAP mais si oui, on pourrait donc effectivement avoir un rapport d'erreur en front : toujours d'abord chercher si le compte existe, déterminer sa méthode d'authentification, et si ça plante lui dire une raison (et pour LDAP "désolé le serveur d'auth est par terre"). On pourrait ainsi être beaucoup plus rusés qu'actuellement et ne pas chercher à contacter de LDAP quand un usager lambda tente de se connecter.

Mais je trouve que la création de compte à la volée, ça rend quand même la connexion avec un LDAP très efficace. Aucun temp d'attente, ça marche immédiatement, lors de la création d'un nouveau compte dans l'annuaire mais aussi quand il change de groupe. Pour moi, on perdrait quelque chose.

#4 - 30 avril 2021 10:19 - Benjamin Renard

Thomas Noël a écrit :

> Benjamin Dauvergne a écrit :
>
> > Comme on garde username/email a priori on sait, en fait je me disais aussi que le provisionning LDAP à la volée était une mauvaise idée de toute façon
> > on ne perd rien ou pas grand chose à ne créer les comptes que dans les synchros.
>
> Mais je trouve que la création de compte à la volée, ça rend quand même la connexion avec un LDAP très efficace. Aucun temp d'attente, ça marche immédiatement, lors de la création d'un nouveau compte dans l'annuaire mais aussi quand il change de groupe. Pour moi, on perdrait quelque chose.

Complètement d'accord avec Thomas : si vous implémenter ça, se serait cool de laisser possible le provisionning à la connexion via un paramètre de configuration (idéal actif par défaut, pour ne pas introduire une perte de fonctionnalité "par défaut").

Et concernant le fait de ne pas loguer les erreurs LDAP, ça me parait un peu violent : les logs servent à de-buguer et sans eux, ça peut-être un enfer. Un utilisateur ne sachant rarement dire qu'elle erreur il a vue s'afficher à l'écran, ça va devenir une source de frustration plus qu'autre chose. À minima, gardez-les en logs debug svp (même si se serait dommage de devoir activer les logs debug en prod pour ça).

#5 - 30 avril 2021 10:44 - Benjamin Dauvergne

Benjamin Renard a écrit :

> Thomas Noël a écrit :
>
> > Benjamin Dauvergne a écrit :
> >
> > > Comme on garde username/email a priori on sait, en fait je me disais aussi que le provisionning LDAP à la volée était une mauvaise idée de toute façon
> > > on ne perd rien ou pas grand chose à ne créer les comptes que dans les synchros.
> >
> > Mais je trouve que la création de compte à la volée, ça rend quand même la connexion avec un LDAP très efficace. Aucun temp d'attente, ça marche immédiatement, lors de la création d'un nouveau compte dans l'annuaire mais aussi quand il change de groupe. Pour moi, on perdrait quelque chose.
>
> Complètement d'accord avec Thomas : si vous implémenter ça, se serait cool de laisser possible le provisionning à la connexion via un paramètre de configuration (idéal actif par défaut, pour ne pas introduire une perte de fonctionnalité "par défaut").

On a quand même des soucis de perf et stabilité actuellement quand on 2 LDAP raccordés un peu lent. L'idéal serait quand même qu'on recherche d'abord par username/email, quand on trouve un utilisateur s'il est du type "authentification LDAP" on regarde directement sur le bon LDAP, uniquement si aucun compte local ne match on bascule sur un mode provisionning, et là aussi idéalement en fonction du username/email (via des motifs, par exemple :

- email ~ '*@monclient.com' -> LDAP du client,
- username ~ */DOMAINE.WINDOWS.DU.CLIENT -> LDAP client)
  on choisit le bon LDAP. Il faut limiter au maximum les cas de recherche exhaustive où on essaie tous les LDAP configurés, et par défaut j'aimerai qu'on ne le fasse plus du tout. Car toute connexion LDAP (pour Publik) ralentit les connexions ratés (mauvais mot de passe, bim ça va essayer aussi sur le LDAP) des gens qui n'ont qu'un compte local voir peut amener des timeout.

Par contre ce qu'il y a c'est que ça sort du modèle d'AuthenticationBackend de Django qui est purement linéaire, j'essaye le 1er, puis le second, etc.. dès qu'un backend répond c'est terminé. La seule exception c'est qu'on peut lever l'exception PermissionDenied à tout moment pour arrêter le processus, mais c'est tout. Il faudrait en fait un seul backend local qui lui même ait une stratégie plus complexe que ce que propose Django basé sur plusieurs backend (qui implémente ce que je dis plus haut).

> Et concernant le fait de ne pas loguer les erreurs LDAP, ça me parait un peu violent : les logs servent à de-buguer et sans eux, ça peut-être un enfer. Un utilisateur ne sachant rarement dire qu'elle erreur il a vue s'afficher à l'écran, ça va devenir une source de frustration plus qu'autre chose. À minima, gardez-les en logs debug svp (même si se serait dommage de devoir activer les logs debug en prod pour ça).

Oui je pense que l'idée de Thomas c'est ne plus le pousser au niveau erreur qui chez nous déclenche l'envoi de traces par mails au monde entier ainsi qu'un ticket dans sentry. Journaliser ces erreurs au niveau warning n'est pas un problème.