django-mellon - Development #55557

Ajouter une vue de debug de l'authentification

13 juillet 2021 11:26 - Valentin Deniaud

Statut:	Fermé	**Début:**	13 juillet 2021
Priorité:	Normal	**Echéance:**	
Assigné à:	Valentin Deniaud	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Benjamin Dauvergne a écrit (#55124) :

> une vue de debug accessible quand on est superadmin, ça reprendrait globalement le code de la vu de login sauf la partie qui pose l'utilisateu ne session, ça ferait un rendu de tout ce qui est obtenu (attributs / assertions) dans une page HTML

Révisions associées

Révision dde8fa5d - 03 août 2021 11:12 - Valentin Deniaud

views: move login code to separate method (#55557)

Révision dbdd6fd7 - 03 août 2021 11:59 - Valentin Deniaud

views: add debug login view (#55557)

Historique

#1 - 13 juillet 2021 11:37 - Valentin Deniaud

- Fichier 0001-wip.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Ça fonctionne mais c'est pas le plus beau patch du monde, je veux bien un avis avant ajout de tests et compagnie.

En todo il y aurait aussi à ajouter un peu de code pour afficher du joli XML, là ça fait une page de 20km de large.

#2 - 13 juillet 2021 11:51 - Frédéric Péters

Je me pose la question de l'utilité, souvent on a à débugguer un accès par un tiers, pas son propre compte admin. Dernier exemple en date, #55353#note-3, où on reste à devoir deviner "Il doit manquer une des clés qui sont indiqués comme mandatory", et ce patch n'aidera pas.

On ne pourrait pas imaginer, authentic, journaliser toutes les assertions (valides/signées) reçues ?

Ainsi on aurait au max à demander à quelle heure la tentative, filtrer le journal vers ce moment, et lire l'assertion.

#3 - 13 juillet 2021 13:14 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Je me pose la question de l'utilité, souvent on a à débugguer un accès par un tiers, pas son propre compte admin. Dernier exemple en date, #55353#note-3, où on reste à devoir deviner "Il doit manquer une des clés qui sont indiqués comme mandatory", et ce patch n'aidera pas.

Il y a déjà un message qui s'affiche dans ce cas; là il n'est pas là, donc ce n'est peut-être pas le problème. L'assertion aurait été poussé dans les logs que ce serait pareil, vu qu'on a plus les logs.

> On ne pourrait pas imaginer, authentic, journaliser toutes les assertions (valides/signées) reçues ?

Oui, mais avec des logs de deux semaines, et deux semaines pour que quelqu'un regarde dans les logs ça n'aidera pas.

#4 - 13 juillet 2021 13:24 - Frédéric Péters

Quand j'écris "journaliser" je pensais au journal, /manage/journal/, il n'y a pas cette expiration à deux semaines là.

#5 - 13 juillet 2021 14:11 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Quand j'écris "journaliser" je pensais au journal, /manage/journal/, il n'y a pas cette expiration à deux semaines là.

Alors là ça devient un autre ticket, il faudrait un logging.Handler à poser sur les domaines authentic2_auth_saml et mellon qui enverrait les messages de auth_saml/mellon dans le journal, je suppose qu'il faudrait aussi ajouter au journal la possibilité d'afficher des popups dans le style des logs passerelle pour les objets un peu gros ne rentrant pas sur une ligne (style un message SAML). Ceci pour éviter d'avoir douzes façons de logger dans le code.

#6 - 13 juillet 2021 14:36 - Valentin Deniaud

Benjamin Dauvergne a écrit :

> Alors là ça devient un autre ticket

Moi je pensais que c'était exclu parce que ce message SAML est gros : encodé en ASCII chez moi ça fait 6Ko, si on se dit qu'on a 5000 connexions par jour par SAML, ça fait 1Go de données par mois. Voilà j'ai aucune idée de si c'est signifiant ou si c'est rien du tout.

> il faudrait un logging.Handler à poser sur les domaines authentic2_auth_saml et mellon qui enverrait les messages de auth_saml/mellon dans le journal

En moins compliqué mais peut-être plus schlag, request.session['mellon_saml_response'] = saml_response et libre à qui utilise mellon d'accéder à ça et d'en faire ce qu'il veut ? C'est deux patches d'une ligne pour arriver à ce qu'on veut.

> je suppose qu'il faudrait aussi ajouter au journal la possibilité d'afficher des popups dans le style des logs passerelle pour les objets un peu gros ne rentrant pas sur une ligne (style un message SAML)

Ça oui mais ça me paraît une bonne évolution.

#7 - 15 juillet 2021 11:19 - Benjamin Dauvergne

Valentin Deniaud a écrit :

> Benjamin Dauvergne a écrit :
>
> > Alors là ça devient un autre ticket
>
> Moi je pensais que c'était exclu parce que ce message SAML est gros : encodé en ASCII chez moi ça fait 6Ko, si on se dit qu'on a 5000 connexions par jour par SAML, ça fait 1Go de données par mois. Voilà j'ai aucune idée de si c'est signifiant ou si c'est rien du tout.

Je n'avais pas fait le calcul en tête, mais oui ça doit être ça, c'est pour ça que je suggérai une vue de debug parce que le cas nominal ce sont les soucis au premier raccordement, où un développeur/CPT est en train de faire le raccordement et donc peut utiliser cette vue de debug.

Le cas lié ici d'une personne pour laquelle ça ne marche pas, alors que ça marche pour les autres, ça devrait être réglé par des logs précis que normalement on a déjà, la résolution du ticket nous éclairera sur ce point. Je ne suis pas certain que dumper l'assertion nous aide plus.

> > il faudrait un logging.Handler à poser sur les domaines authentic2_auth_saml et mellon qui enverrait les messages de auth_saml/mellon dans le journal
>
> En moins compliqué mais peut-être plus schlag, request.session['mellon_saml_response'] = saml_response et libre à qui utilise mellon d'accéder à ça et d'en faire ce qu'il veut ? C'est deux patches d'une ligne pour arriver à ce qu'on veut.

Oui aussi.

> > je suppose qu'il faudrait aussi ajouter au journal la possibilité d'afficher des popups dans le style des logs passerelle pour les objets un peu gros ne rentrant pas sur une ligne (style un message SAML)
>
> Ça oui mais ça me paraît une bonne évolution.

Oui mais avant de se disperser je me dis qu'il faut comprendre pourquoi dans le cas lié on a pas le message d'erreur en front avec l'erreur de mapping qu'on aurait du avoir si c'est vraiment une erreur de mapping (manque un attribut) qui a bloqué le SSO.

#8 - 15 juillet 2021 11:31 - Benjamin Dauvergne

Je reviens au patch, plutôt que d'utiliser lasso_node_dump() utiliser lasso_node_debug(), via response_dump = login.response and login.response.debug(4) (4 c'est l'indentation), je dumperai login.assertion aussi (c'est dans la réponse, mais c'est plus court à analyser).

Et les choses dans cet ordre, de plus utile au moins utile :

- attributs
- assertion
- response
- artifact/POST (parce que msgBody ça peut aussi être le contenu du POST en cas de binding POST, en fait ce sera la même chose que dans login.response mais encodé).

Est-ce qu'on ne laisserait pas la vue active pour tout le monde aussi si DEBUG = True ?

Et si le cas DEBUG = True est implémenté: j'ai l'impression qu'une fois mellon_debug_login posé il ne se ré-initialise jamais, il faudrait le virer dans DebugLoginView.authenticate, et ajouter un lien "Try Again" au template (Try again).

#9 - 15 juillet 2021 12:38 - Benjamin Dauvergne

Et dernière remarque : je laissera l'authentification se faire (via django.contrib.auth.authenticate()), juste je ne loggerai pas l'utilisateur (on shunte l'appel à utils.login(...)), et j'afficherai l'utilisateur qui serait authentifié ainsi que tous les logs entre temps (via https://docs.python.org/3/library/logging.handlers.html#memoryhandler, posé sur le logger mellon et authentic2_auth_saml temporairement). Comme ça on sait tout.

PS: aussi exemple, https://docs.python.org/3/howto/logging-cookbook.html#buffering-logging-messages-and-outputting-them-conditionally

#10 - 19 juillet 2021 17:04 - Valentin Deniaud

- Fichier 0001-wip.patch ajouté

Voilà j'ai implémenté tes idées, j'attends quand même une dernière approbation avant de fignoler et d'écrire les tests.

Benjamin Dauvergne a écrit :

> tous les logs entre temps (via https://docs.python.org/3/library/logging.handlers.html#memoryhandler, posé sur le logger mellon et authentic2_auth_saml temporairement). Comme ça on sait tout.

Je pense qu'on a pas le droit de faire référence à authentic2_auth_saml mais poser le handler sur le root logger ça fait ce qu'on veut j'ai l'impression. Aussi pas compris l'histoire de MemoryHandler, j'ai fait sans.

#11 - 29 juillet 2021 11:46 - Thomas Noël

Selon moi "if not request.user.is_superuser" est en trop, ça va compliquer le debug quand on n'aura pas un accès à l'idp (pas de compte test ou pas d'accès à l'idp du tout : mon idée est est dans ce cas on inviterait le client à tester par lui même et nous envoyer la page de résultat en copier-coller).

Aussi, y'a un self.log.root.addHandler, mais pas de suppression ensuite : est-ce qu'on ne risque pas ici d'activer un log permanent par la suite ? (Si c'est trop compliqué, je pense qu'on peut retirer les logs dans un premier temps, histoire d'avoir au moins un debug SAML de base)

#12 - 29 juillet 2021 12:28 - Benjamin Dauvergne

Valentin Deniaud:

> Je pense qu'on a pas le droit de faire référence à authentic2_auth_saml mais poser le handler sur le root logger ça fait ce qu'on veut j'ai l'impression. Aussi pas compris l'histoire de MemoryHandler, j'ai fait sans.

Parce qu'à la lecture de la doc de logging je me suis laissé à penser qu'il fallait faire absolument comme ça, mais ton StreamHandler sur un io.StreamIO semble effectivement encore plus simple et naturel.

Thomas Noël a écrit :

> Selon moi "if not request.user.is_superuser" est en trop, ça va compliquer le debug quand on n'aura pas un accès à l'idp (pas de compte test ou pas d'accès à l'idp du tout : mon idée est est dans ce cas on inviterait le client à tester par lui même et nous envoyer la page de résultat en copier-coller).

Ça me gêne vraiment que ce soit ouvert tout le temps (les audits à la con tout ça), mais si superuser est jugé trop dur, on peut se baser sur settings.DEBUG; pour l'utiliser en prod pour nous ou un client il suffit de mettre le ou les IPs concernée dans les IPs de debug dans hobo (/debug/) sur hobo.

> Aussi, y'a un self.log.root.addHandler, mais pas de suppression ensuite : est-ce qu'on ne risque pas ici d'activer un log permanent par la suite ?

> (Si c'est trop compliqué, je pense qu'on peut retirer les logs dans un premier temps, histoire d'avoir au moins un debug SAML de base)

Oui pareil pour addHandler j'ai fait une remarque équivalente sur un addFilter sur un ticket passerelle, faut faire attention quand on touche aux loggers, sinon on laisse du bordel derrière, j'en ferai un contextmanager comme je l'ai dit dans l'autre ticket, c'est le code pythonesque normal pour allouer/désallouer une ressource.

#13 - 29 juillet 2021 14:12 - Thomas Noël

Benjamin Dauvergne a écrit :

> Thomas Noël a écrit :
>
> > Selon moi "if not request.user.is_superuser" est en trop, ça va compliquer le debug quand on n'aura pas un accès à l'idp (pas de compte test ou pas d'accès à l'idp du tout : mon idée est est dans ce cas on inviterait le client à tester par lui même et nous envoyer la page de résultat en copier-coller).
>
> Ça me gêne vraiment que ce soit ouvert tout le temps (les audits à la con tout ça), mais si superuser est jugé trop dur, on peut se baser sur settings.DEBUG; pour l'utiliser en prod pour nous ou un client il suffit de mettre le ou les IPs concernée dans les IPs de debug dans hobo (/debug/) sur hobo.

Ouaip mais bon, DEBUG=True ça pourrait suffire, selon moi (ça s'expliquer lors d'un audit)... Dans le wip proposé c'est « if not request.user.is_superuser or not settings.DEBUG: » et je proposais de remplacer par « if not settings.DEBUG: »...

Ceci dit, ça ferme le debug en prod (on n'y fait jamais DEBUG=True), et donc, pourquoi pas : « if not (settings.DEBUG or request.user.is_superuser): » ?

#14 - 29 juillet 2021 16:51 - Benjamin Dauvergne

Thomas Noël a écrit :

> Ceci dit, ça ferme le debug en prod (on n'y fait jamais DEBUG=True), et donc, pourquoi pas : « if not (settings.DEBUG or request.user.is_superuser): » ?

On peut toujours activer le DEBUG en prod via https://hobo.client.fr/debug/ pour des adresses IPs données.

#15 - 03 août 2021 12:05 - Valentin Deniaud

- Fichier 0001-views-move-login-code-to-separate-method-55557.patch ajouté

- Fichier 0002-views-add-debug-login-view-55557.patch ajouté

Version relisable, vous avez l'air d'accord pour se baser uniquement sur DEBUG alors j'ai fait ça.

J'ai complètement revu la logique de l'ensemble, maintenant DebugLoginView c'est une vue simple qui pose un cookie et LoginView affiche les infos de debug en fonction.

#16 - 03 août 2021 15:52 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

#17 - 03 août 2021 16:00 - Valentin Deniaud

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

```
commit dbdd6fd70b26bc32cdfbd4917f5a0bf4fafaa276
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date:   Tue Aug 3 11:15:57 2021 +0200

    views: add debug login view (#55557)

commit dde8fa5d026f69a0bd1ae41468741412f901816b
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date:   Tue Aug 3 11:12:18 2021 +0200

    views: move login code to separate method (#55557)
```

#18 - 05 août 2021 23:18 - Frédéric Péters

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

Fichiers

0001-wip.patch	3,3 ko	13 juillet 2021	Valentin Deniaud
0001-wip.patch	6,97 ko	19 juillet 2021	Valentin Deniaud
0001-views-move-login-code-to-separate-method-55557.patch	2,98 ko	03 août 2021	Valentin Deniaud
0002-views-add-debug-login-view-55557.patch	6,9 ko	03 août 2021	Valentin Deniaud