EoleSSO - Tâche #29770

Scénario # 29670 (Terminé (Sprint)): Proposer nativement EoleSSO sur les ports 8443 et 443

Étude de la demande

24/03/2020 07:55 - Emmanuel GARETTE

Statut:	Fermé	**Début:**	24/03/2020
Priorité:	Normal	**Echéance:**	
Assigné à:	Emmanuel GARETTE	**% réalisé:**	100%
Version cible:	Prestation Cadoles MEN 13-15	**Temps estimé:**	0.00 heure
		Temps passé:	0.00 heure
Description			

Révisions associées

Révision 5ff6200b - 27/03/2020 14:51 - Emmanuel GARETTE

le port par défaut de eolesso est 443 en 2.8.0 (ref #29770)

Historique

#1 - 24/03/2020 07:55 - Emmanuel GARETTE

- Statut changé de Nouveau à En cours

#2 - 24/03/2020 08:17 - Emmanuel GARETTE

eolesso_port

le serveur SSO est distant

La valeur par défaut reste 8443 en 2.7
La valeur par défaut est 443 en 2.8.0

le serveur SSO est local

la valeur n'est pas 8443

Faire écouter le service python sur ce port.
Le client utilise ce port.

Le service n'est pas utilisable sur le port 443 (puisque pas configurable sur ce port).

la valeur est 8443

C'est la valeur par défaut sur 2.7.1.

Le service écoute sur le port 8443 (il n'y a pas de reverse proxy entre l'utilisateur et le service).
Le service est accessible sur le port 443/%%eolesso_cas_folder :

- si apache est activé, le proxy est géré par apache
- sinon utilise nginx

la valeur est 443

C'est la valeur par défaut sur 2.8.0.

Le service écoute sur le port 8443 (il n'y a pas de reverse proxy entre l'utilisateur et le service).
Le service est accessible sur le port 443/%%eolesso_cas_folder :

- si apache est activé, le proxy est géré par apache
- sinon utilise nginx

eolesso_cas_folder

La variable passe en mode normal (mode expert aujourd'hui).

le serveur SSO est distant

La valeur par défaut de %%eolesso_cas_folder est "".

le serveur SSO est local

la valeur n'est pas 8443

La valeur par défaut de %%eolesso_cas_folder est "".

la valeur est 443

2.7.1

La valeur par défaut de %%eolesso_cas_folder est "".
L'utilisateur devra renseigner "/sso" pour que cela fonctionne dans cette configuration.

2.8.0

La valeur par défaut de %%eolesso_cas_folder est "/sso".

A propos de la variable eolesso_css

Attention il faut un nom de fichier relatif par rapport à la racine du site, /monfichier.css ne fonctionnera que sur le port 8443. /sso/monfichier.css ne fonctionnera que sur le port 443. monfichier.css fonctionnera sur les 2 ports.

#3 - 24/03/2020 08:33 - Emmanuel GARETTE

- Statut changé de En cours à Résolu

- % réalisé changé de 0 à 100

#4 - 26/03/2020 08:41 - Emmanuel GARETTE

Je ne vois pas comment implémenter ce qui est demandé de manière sécurisé.

Pour que cela fonctionne il faut que le cookie soit valable sur /. Cela signifie que n'importe quelle application local peut avoir accès au cookie. Une faille XSS quelconque dans une application et un attaquant peut récupérer une session.

Il faudrait limiter le cookie sur / sur le port 8443 et sur /sso sur le port 443. Malheureusement cela n'est pas possible.

Il faudrait donc que eolesso écoute sur https://IP:8443/sso et non https://IP:8443/. Cela veut dire qu'on perd la compatibilité avec les applications existantes.
On peut imaginer de faire un "redirect" mais comment être sur que toutes les applications supportent le redirect ? (je pense à cas-sso, imapd, ...).

On ne peut pas faire ces changements tout en garantissant qu'il n'y ait pas de régression.

Je propose une nouvelle solution :

Le service n'écoute que sur un port.

Si le port est 443, on met /sso dans le cookie, sinon on laisse "/".

#5 - 26/03/2020 11:38 - Emmanuel GARETTE

Je m'aperçois que les virtualhosts ne sont pas correctement gérer sur Scribe.

Dans le schéma suivant :

```
                 > SSO \  scribe.ac-test.fr
-------------- /         >    ----------
| navigateur |                | Scribe |
-------------- \          >    ----------
                > Envole/   scribe.ac-test.fr
```

Il n'y a pas de différence.

Ni dans le schéma suivant :

```
                 > SSO \  scribe.ac-test.fr
-------------- /         >    ----------        ----------
| navigateur |                | Amon   | -----> | Scribe |
-------------- \          >    ----------        ----------
                > Envole/   scribe.ac-test.fr
```

Par contre il y a une différence dans le schéma suivant :

```
                 > SSO \  sso.scribe.ac-test.fr
-------------- /         >    ----------        ----------
| navigateur |                | Amon   | -----> | Scribe |
-------------- \          >    ----------        ----------
                > Envole/   scribe.ac-test.fr
```

C'est le cas courant qu'on utilise à Cadoles.

Dans ce cas, le SSO n'est accessible que sur le nom de domaine "sso.scribe.ac-test.fr". Maintenant il sera également accessible sur le domaine "scribe.ac-test.fr".

#6 - 26/03/2020 11:50 - Daniel Dehennin

Je vais essayer de résumer ce que je comprends et ce que je voyais pour ce scénario.

Accès SSO par le port 8443 par les clients

Rien ne change pour les clients, ils y accèdent avec l'URL **https://mon-etablissement.example.fr:8443**/

Le cookie est dédié à cette URL

Accès SSO par le port 443 par les clients

L'accès par les clients au serveur peut se faire :

- sur un sous répertoire comme **https://mon-etablissement.example.fr/sso** s'il y a d'autres applications à côté afin
  - d'avoir la racine pour une redirection par défaut (**roundcube** sur scribe)
  - d'éviter le vol de session par une autre application car le cookie serait valable pour toutes les applications sous la racine
- sur la racine **https://mon-serveur-sso-dédié.example.fr**/ (ce qui rejoint la configuration dont parle Emmanuel) si
  - le serveur SSO n'a pas d'autres applications
  - le serveur SSO à un nom DNS dédié (dans ce cas le cookie ne sera valable que pour ce nom)

Configuration du serveur SSO pour les clients

L'idée de base était de faire une configuration classique de serveur d'application :

- le serveur en python n'écoute que sur **127.0.0.1:8443**
- seul un reverse proxy expose le SSO sur le nom de domaine déclaré du SSO

Voilà une idée de configuration pour nginx en example :

```
upstream eolesso {
     server localhost:8443;
}

server {
    listen         XX.XX.XX.XX:443;
    server_name  mon-scribe.example.fr;

    access_log  /var/log/nginx/scribe-access.log;
    error_log   /var/log/nginx/scribe-error.log;

        ssl_protocols  TLSv1.2 TLSv1.3;

        ssl_certificate /etc/ssl/certs/my-server.pem;
        ssl_certificate_key /etc/ssl/private/my-server.key;

    location /sso {
        root /chemin/des/fichiers/sso/static;
        try_files $uri @eolesso;
    }

    location @eolesso {
         include proxy_params;
         proxy_pass https://eolesso;
    }
}
server {
    # Compatibility with old clients using 8443
    listen         XX.XX.XX.XX:8443;
    server_name  mon-scribe.example.fr;

    access_log  /var/log/nginx/eolesso-access.log;
    error_log   /var/log/nginx/eolesso-error.log;

        ssl_protocols  TLSv1.2 TLSv1.3;

        ssl_certificate /etc/ssl/certs/my-server.pem;
        ssl_certificate_key /etc/ssl/private/my-server.key;

    location / {
        root /chemin/des/fichiers/sso/static;
        try_files $uri @eolesso;
    }
```

```
    location @eolesso {
         include proxy_params;
         proxy_pass https://eolesso;
    }
}
```

#7 - 26/03/2020 12:05 - Daniel Dehennin

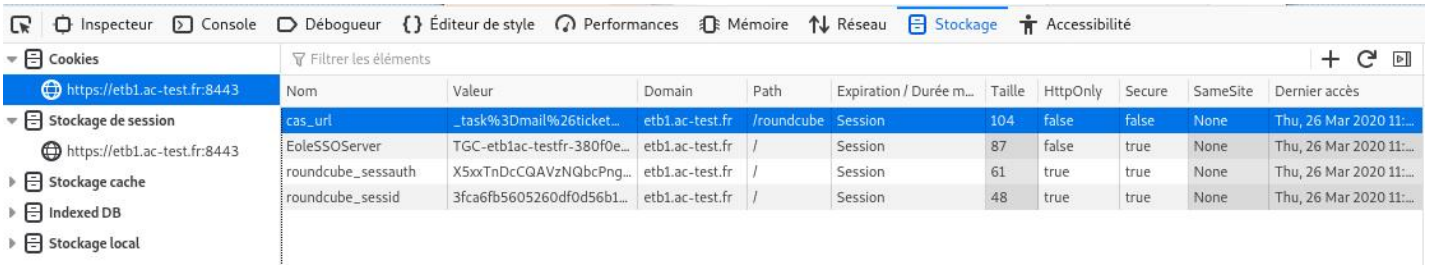
- Fichier cookie-8443.png ajouté

Correction

Le port d'écoute du serveur n'est pas pris en compte dans la restriction de cookie.

Dans le cas d'applications multiples sur le même nom DNS que le SSO, il semble impératif d'avoir un sous répertoire pour réstreindre ce cookie, par example **https://mon-établissement.example.fr/sso**.

Sur un **etb1**, après authentification, j'obtiens le cookie avec les valeurs suivantes:



#8 - 06/04/2020 12:39 - Joël Cuissinat

- Statut changé de Résolu à Fermé

- Restant à faire (heures) mis à 0.0

Fichiers

cookie-8443.png	57,6 ko	26/03/2020	Daniel Dehennin