Sécuriser l'accès à kibana

Principe

Pour limiter l'accès aux données contenues dans Agimus-NG, nous allons utiliser le serveur hébergeant kibana pour restreindre l'accès à elasticsearch.

Suivant vos choix de déploiement, il peut y avoir des variations mais nous allons supposer que vous utilisez 3 serveurs (au moins, ça peut être plus si vous avez un cluster pour elasticsearch) dans le cadre d'agimus :

- le **frontal** (indicateur.univ.fr)
- **kibana** (agimus.univ.fr) qui héberge une ou plusieurs instances de kibana et va servir de proxy
- **elasticsearch** (es-agimus.univ.fr)

Évidemment, il est possible que votre instance elasticsearch soit sur le même serveur que kibana ou que le serveur elasticsearch soit en réalité un cluster. Nous exposons ici une proposition de mise en place, si vous rencontrez des difficultés pour l'adapter dans votre environnement n'hésitez pas à poser vos question sur la liste esup-utilisateurs@esup-portail.org

Frontal

Le frontal ne fait qu'afficher une interface réduite de kibana sous forme d'iframe. Elle permet de gérer les droits d'accès aux différents graphiques suivant des profils et d'empêcher un utilisateur de facilement modifier les graphiques créés. Pour retrouver le code de ce frontal, rendez-vous à l'adresse suivante :

https://github.com/EsupPortail/agimus-ng/tree/master/frontal

> Si quelqu'un a accès via le frontal, il a également accès à kibana et peut, s'il connaît l'url, modifier les graphiques créés. C'est pourquoi nous vous conseillons de sauvegarder régulièrement vos configurations kibana avec les scripts proposés sur le dépôt github.

Fonctionnalité d'export des tableaux de bord

Une nouvelle fonctionnalité du frontal consiste à rendre possible l'export d'un tableau de bord par les utilisateurs. Pour l'activer suivez la documentation Ajouter la fonctionnalité d'export au frontal.

Précautions à prendre

Les tableaux de bord rendus disponibles dans le frontal ne doivent pas contenir de date pré-enregistées sinon le sélecteur de date du frontal ne fonctionnera pas correctement.

L'ajout d'un nouveau groupe d'utilisateur nécessite l'ajout dans le fichier de configuration pour pouvoir être utilisé dans l'interface d'administration.

Kibana

Le serveur hébergeant kibana va servir de proxy pour l'accès à elasticsearch.

Lorsque vous affichez des graphiques kibana, ce n'est pas le navigateur client mais kibana qui va interroger le cluster elasticsearch. Il est donc possible de limiter l'accès en utilisant la machine comme proxy qui filtrera l'accès au cluster elasticsearch mais également, sur d'autres critères, aux graphiques générés par kibana.

Pour cela nous utilisons dans notre exemple mod_cas qui va nous permettre de limiter l'accès par cassification.

Elasticsearch

Le serveur (ou cluster) ne sera accessible que par la machine kibana et celle effectuant les traitements logstash.

Mise en pratique

Configuration du frontal

Paramétrage de gestion des groupes pour le frontal (front-agimus-ng/config/config.php)

```
    $user_roles = array('ROLE_ADMIN', 'ROLE_MANAGER', 'ROLE_USER');
    $user_roles_mapping = array(
      "ROLE_ADMIN" => "nom_du_groupe_ldap_ADMIN",
      "ROLE_MANAGER" => "nom_du_groupe_ldap_MANAGER",
      "ROLE_USER" => "nom_du_groupe_ldap_USER"
    );
```

Nous vous proposons pour le moment une configuration avec apache mais n'hésitez à nous transmettre d'autres propositions pour compléter

Apache

La configuration au niveau du frontal n'a rien de particulier. Il s'agit simplement de déployer l'application php disponible sur le dépôt github : https://github.com/EsupPortail/agimus-ng/tree/master/experimentation/front-agimus-ng

La configuration ci-dessous est celle du serveur hébergeant kibana. On présente le cas où le cluster contient 2 serveurs et le cas où le serveur est sur la même machine.

Configuration apache sur le serveur kibana

```
RewriteEngine on
RewriteRule "^/$" "/kibana/"  [R]

CustomLog logs/access.agimus.log combined
ErrorLog logs/error.agimus.log

RemoteIPHeader X-Forwarded-For

# Configuration du module mod_cas
include conf.modules.d/00-cas.conf

<Proxy *>
    Order allow,deny
    Allow from all
</Proxy>

<Location "/kibana/">
    <RequireAny>
                # Autorisation d'appel du serveur kibana sur lui même sans authentification (123.456.789.012)
        Require ip 123.456.789.012 127.0.0.1
                # Autorisation des comptes CAS
        AuthType CAS
        CASScope /

        #### 2 types d'autorisation possibles
        ### On utilise uniquement le CAS et une autorisation par utilisateurs
        require user login1 login2 login3

        ### On autorise en se basant sur un attribut LDAP qui peut également être utilisé dans la configuration
du frontal
        # AuthLDAPURL "ldap://ldap.univ.fr:389/ou=people,dc=univ,dc=fr?uid??"
        # AuthLDAPBindDN "cn=app-agimus,ou=system,dc=univ,dc=fr"
        # AuthLDAPBindPassword "XXXXXXXXX"
        ## Groupe englobant l'ensemble des groupes utilisés dans le frontal
        # require ldap-filter group=Agimus-user

    </RequireAny>
    ProxyPass http://localhost:5601/
    ProxyPassReverse http://localhost:5601/
```

```
</Location>

###############
# Cas où elasticsearch est sur le même serveur que kibana
###############
#<Location "/elasticsearch">
#    <RequireAny>
#        # Autorisation d'appel du serveur kibana sur lui même sans authentification (123.456.789.012)
#        Require ip 123.456.789.012 127.0.0.1
#                # Autorisation des comptes CAS (ici on limite aux utilisateurs suivant le cluster, les
informaticiens)
#        AuthType CAS
#        CASScope /
#        require user login1
#    </RequireAny>
#    ProxyPass http://localhost:9200/
#    ProxyPassReverse http://localhost:9200/
#</Location>
#
################


###############
# Cas où elasticsearch est un cluster
###############

<Proxy balancer://EScluster>
BalancerMember http://es1.univ.fr:9200 route=es1
BalancerMember http://es2.univ.fr:9200 route=es2
</Proxy>

Header add Set-Cookie "AgimusRoute=route.%{BALANCER_WORKER_ROUTE}e; path=/" env=BALANCER_ROUTE_CHANGED

<Location "/elasticsearch">
    <RequireAny>
        # Autorisation d'appel du serveur kibana sur lui même sans authentification (123.456.789.012)
        Require ip 123.456.789.012
                Require local
                # Autorisation des comptes CAS (ici on limite aux utilisateurs suivant le cluster, les
informaticiens)
        AuthType CAS
        CASScope /
        require user login1
    </RequireAny>
    ProxyPass balancer://EScluster stickysession=AgimusRoute
    ProxyPassReverse balancer://EScluster
</Location>

ProxyPreserveHost on
#Empêche le proxy apache (nous on ne fait que du reverse proxy)
ProxyRequests Off
```

Si vous ne l'utilisez pas déjà, il vous faudra ajouter le module mod_auth_cas : https://github.com/Jasig/mod_auth_cas et le paramétrer grâce au fichier de configuration ci-dessous qui est appelé dans le fichier de configuration ci-dessus.

Configuration apache du module mod_cas

```
LoadModule auth_cas_module modulesUL/mod_auth_cas.so

CASVersion 2
CASDebug Off
CASLoginURL https://cas.univ.fr
CASValidateURL https://cas.univ.fr/serviceValidate
CASCookiePath /var/cache/httpd/cas/
CASCookieHttpOnly Off
```