Sécurisation d'un cluster Apache Solr sous CentOS 7

29 août 2018

1 Objectifs

Par défaut, Apache Solr ne vérifie pas l'identité des utilisateurs avant de les laisser accéder aux ressources du cluster. Apache Solr propose cependant des outils permettant de sécuriser le cluster. Ces outils prennent en compte l'authentification et l'authorisation des utilisateurs. Ce qui permet la vérification de l'identité des utilisateurs et la restriction de l'accès aux ressources du cluster Solr. Dans ce document, nous utiliserons l'un de ces outils pour sécuriser notre cluster Solr.

2 Configuration de l'authentfication et de l'autorisation

Le fichier principal qui contient les informations d'authentification et d'authorisation est nommée "`security.json`". Ce fichier doit être placé dans le répertoire "`SOLR_HOME`" si Solr n'est pas utilisé en mode cluster. Par contre, si Solr est utilisé en mode SolrCloud, ce fichier doit être tranféré dans l'ensemble Zookeeper chargé de gérer les serveurs Solr.

La commande ci-dessous créee un fichier "`security.json`" dans lequel sont indiqués les modes d'authentification et d'autorisation ainsi qu'un compte utilisateur nommé "`solr`" avec pour mot de passe "`SolrRocks`" (dont la valeur hashée est présentée) qui a les droits d'administration sur les paramètres de sécurité :

```
# cat > /opt/solr-7.4.0/server/solr/security.json << "EOF"
{
"authentication":{
   "blockUnknown": true,
   "class":"solr.BasicAuthPlugin",
   "credentials":{"solr":"IV0EHq1OnNrj6gvRCwvFwTrZ1+z1oBbnQdiVC3otuq0= Ndd7LKvVBAaZIF0QAV
   i1ekCfAJXr1GGfLtRUXhgrF8c="}
},
"authorization":{
   "class":"solr.RuleBasedAuthorizationPlugin",
   "permissions":[{"name":"security-edit",
      "role":"admin"}],
   "user-role":{"solr":"admin"}
```

```
}}
EOF
```

Il est important de noter que l'attribution de la valeur "`true`" au paramètre "`blockUnknown`" dans le fichier "`security.json`" créée ci-dessus est crucial, car c'est ce dernier qui autho-rise Solr a vérifier l'identité des utilisateurs avant de les permettre d'accéder au cluster. En effet, la valeur "`false`" est attribuée par défaut à ce paramètre.

On peut optionnellement permettre à la ligne de commande "`$ solr`" d'exploiter au-tomatiquement les informations indéquées dans le fichier de base "`security.json`" en édi-tant dans chaque serveur Solr le fichier "`solr.in.sh`" et en lui approtant les modification ci-dessosus.

```
# vi /opt/solr-7.4.0/bin/solr.in.sh

SOLR_AUTH_TYPE="basic"
SOLR_AUTHENTICATION_OPTS="-Dbasicauth=solr:SolrRocks"
```

Nous chargeons le fichier "`security.json`" dans l'ensemble Zookeeper qui gère notre SolrCloud avec la commande suivante :

```
# solr zk cp file:/opt/solr-7.4.0/server/solr/security.json  zk:/security.json
```

On peut à présent consulter les informations de sécurité en éxécutant dans n'importe quel serveur Solr les deux commandes ci-dessous :

```
# curl --user solr:SolrRocks http://localhost:8983/solr/admin/authentication

# curl --user solr:SolrRocks http://localhost:8983/solr/admin/authorization
```

3 Création et gestion des utilisateurs et des privilèges

La commande ci-dessous utilise le compte utilisateur "`solr`" précédemment créé pour se connecter au cluster Solr et y créer trois autres comptes utilisateurs ("`manager`", "`writer`" and "`reader`") en associant un mot de passe à chacun d'eux :

```
# curl  --user solr:SolrRocks  http://localhost:8983/solr/admin/authentication -
H 'Content-type:application/json'
-d '{"set-user": {"manager":"manager*pw!", "writer":"writer*pw!", "reader":"reader*pw!"}}
```

La commande ci-dessous permet d'attribuer des roles à quelques permissions prédéfinies sur les ressources du cluster :

```
# curl --user solr:SolrRocks -H 'Content-type:application/json' -d '{
  "set-permission": {"name": "security-edit", "role":"manager_security"},
  "set-permission": {"name": "collection-admin-edit", "role":"manager_collection"},
```

```
  "set-permission": {"name": "core-admin-edit", "role":"manager_core"},
  "set-permission": {"name": "update", "role":"writer_update"},
  "set-permission": {"name": "schema-edit", "role":"writer_schema"},
  "set-permission": {"name": "config-edit", "role":"writer_config"},
  "set-permission": {"name": "read", "role":"reader_select"},
  "set-permission": {"name": "schema-read", "role":"reader_schema"},
  "set-permission": {"name": "config-read", "role":"reader_config"}
}' http://localhost:8983/solr/admin/authorization
```

La commande ci-dessous attribue quelques roles précédemment définis aux différents utilisateurs

```
# curl --user solr:SolrRocks -H 'Content-type:application/json' -d '{
   "set-user-role" :
   {"solr": ["manager_security","manager_collection","manager_core","reader_select"],
   "manager": ["manager_security","manager_collection","manager_core","reader_select"],
   "writer": ["writer_update","writer_schema","writer_config","reader_select"],
   "reader": ["reader_select","reader_schema","reader_config"]}
}' http://localhost:8983/solr/admin/authorization
```

Les utilisateurs "`solr`" et "`managers`" ont pratiquement tous les droits. L'utilisateur "`writer`" possède les droits d'écriture et de lecture sur les données indexées dans Solr incluant leurs schémas contrairement à l'utilsateur "`reader`" qui ne possède que le droit de lecture sur ces mêmes données.

On peut consulter les nouvelles informations de sécurité en éxécutant dans n'importe quel serveur Solr les deux commandes ci-dessous :

```
# curl --user manager:manager*pw! http://localhost:8983/solr/admin/authentication

# curl --user manager:manager*pw! http://localhost:8983/solr/admin/authorization
```

On pourra noter sur les résultats de la première commande que tous les mots de passes associés aux comptes utilisateurs apparaissent hashés. On peut se servir de l'une de ces informations pour changer le compte "`solr`" avec son mot de passe "`SolrRocks`" éventuellement dans les deux fichiers "`security.json`" et "`solr.in.sh`". Puis, reprendre la procédure de sécurisation de Solr avec le nouveau compte "`admin`" choisi. En effet, le compte "`solr`" avec son fameux mot de passe "`SolrCloud`" sont bien connus des utilisateurs de Solr. Ce compte est régulièrement exploité à titre indicatif pour illustrer la procédure de sécurisation.

4 Conclusion

Dans ce document nous avons décrit une procédure permettant de configurer la sécurité dans un cluster Solr. Ensuite, nous avons présenté et illustré avec quelques exemples la démarche à suivre pour créer des utilisateurs et pour leur accorder certains privilèges sur les

ressources ou objets de ce cluster. Il vaut la peine de faire remarquer que la sécurisation de Solr repose sur un fichier "`security.json`". Il faut par conséquent prendre le soin de protéger ce fichier en écriture pour ne pas permettre à tout utilisateur malveillant de modifier son contenu. Notons également qu'avec ce mode de protection, des précautions supplémentaires nécessitent d'être prises en compte. On peut suivre le lien suivant pour plus d'informations "`https://lucene.apache.org/solr/guide/7_4/securing-solr.html`". Dans le prochain document, nous allons indexer et consulter de différentes manières les données dans notre cluster Solr. Il s'agira des données issues de quelques expérimentations du SIDEx.

Références