AAF Integration with APPC

Overview

APPC uses the the AAF Shiro OSGI plugin to secure access to ODL web services with AAF.

The AAF shiro plugin is preloaded in the APPC docker image along with a sample cadi.properties file.

Enabling AAF security for APPC using two way certificate

New certificates are available on the master branch to replace expired one way ssl

Heat and other non OOM deployments

1. Use the files in {https://gerrit.onap.org/r/50963}
    a. copy new certificate files into deployment
       /opt/onap/appc/data/storer
       org.onap.appc.keyfile
       org.onap.appc.p12
       truststoreONAPall.jks

    b. copy new cadi.properites file
       /opt/onap/appc/data/properties/cadi.properties
2. edit aaa-aap-config.xml

   /opt/opendaylight/current/etc/opendaylight/datastore/initial/config/aaa-app-config.xml

    a. swap commenting for tokenAuthRealm

```
<main>

  <pair-key>tokenAuthRealm</pair-key>

  <pair-value>org.opendaylight.aaa.shiro.realm.TokenAuthRealm</pair-value>

  <!--      <pair-value>org.onap.aaf.cadi.shiro.AAFRealm</pair-value> -->

</main>
```

To

```
<main>

  <pair-key>tokenAuthRealm</pair-key>

  <!--      <pair-value>org.opendaylight.aaa.shiro.realm.TokenAuthRealm</pair-value> -->

  <pair-value>org.onap.aaf.cadi.shiro.AAFRealm</pair-value>

</main>
```

    b. swap urls for urls to be secured by AAF. NOTE: DO THIS FOR ALL URLS USING authcBasic

```
<urls>

  <pair-key>/**</pair-key>

  <pair-value>authcBasic, roles[admin]</pair-value>

  <!--      <pair-value>authcBasic, roles[org.onap.appc.odl|odl-api|*]</pair-value> -->

</urls>
```

To

```
<urls>

  <pair-key>/**</pair-key>

  <!--      <pair-value>authcBasic, roles[admin]</pair-value> -->

  <pair-value>authcBasic, roles[org.onap.appc.odl|odl-api|*]</pair-value>
```

```
</urls>
```

3. Restart APPC

If there is not a DNS entry for aaf-onap-beijing-test.osaaf.org set the mapping to a valid AAF instance in etc/hosts.

Enabling AAF security for APPC old certificates

1. update cadi.properties with the correct information for your environment.
   /opt/onap/appc/data/properties/cadi.properties

   properties include:

   hostname= usually machine hostname, should be unique

   aaf_url= AAF instance to connect to

   aaf_id= id used to connect to AAF

   aaf_password= password associated with aaf_id

   cadi_keyfile= keyfile used for password encryption
2. edit aaa-aap-config.xml
   /opt/opendaylight/current/etc/opendaylight/datastore/initial/config/aaa-app-config.xml

   a. swap commenting for tokenAuthRealm

```
<main>
    <pair-key>tokenAuthRealm</pair-key>
    <pair-value>org.opendaylight.aaa.shiro.realm.TokenAuthRealm</pair-value>
    <!--        <pair-value>org.onap.aaf.cadi.shiro.AAFRealm</pair-value> -->
</main>
```

To

```
<main>
    <pair-key>tokenAuthRealm</pair-key>
    <!--        <pair-value>org.opendaylight.aaa.shiro.realm.TokenAuthRealm</pair-value> -->
    <pair-value>org.onap.aaf.cadi.shiro.AAFRealm</pair-value>
</main>
```

   b. swap urls for urls to be secured by AAF. NOTE: DO THIS FOR ALL URLS USING authcBasic

```
<urls>
    <pair-key>/**</pair-key>
    <pair-value>authcBasic, roles[admin]</pair-value>
    <!--        <pair-value>authcBasic, roles[org.onap.appc.odl|odl-api|*]</pair-value> -->
</urls>
```

To

```
<urls>
    <pair-key>/**</pair-key>
    <!--        <pair-value>authcBasic, roles[admin]</pair-value> -->
    <pair-value>authcBasic, roles[org.onap.appc.odl|odl-api|*]</pair-value>
</urls>
```

Customization

The permissions used to secure urls can be customized.

To customize the permission used for a url:

1. Ensure the permision has been added to AAF
2. Identify the url in the aaa-app-config.xml
3. set the AAF permission to be used in the roles[] for the url
4. Example:

   to use the permission org.onap.appc.admin|*|* for the /auth/** url

```
 <urls>

   <pair-key>/auth/**</pair-key>

   <pair-value>authcBasic, roles[org.onap.appc.admin|*|*]</pair-value>

</urls>
```

Older ODL versions

Older versions of ODL use shiro.ini located in the /etc directory in place of aaa-app-config.xml. The properties used in shiro.ini are the same. When updating the shiro.ini ODL has to be restared for changes to take effect.

Bath legacy credential support capability

The bath legacy credential support capability allows the legacy admin credentials to still function when AAF is enabled. The legacy admin credentials are stored in the /opt/onap/appc/data/properties/bath-config.csv file. If additional legacy credentials need to be added, they should be in the format expected below with a legacy base 64 encoded login/pw, AAF base 64 encoded login/pw, and expiration date in YYYY-MM-DD format:

Basic <legacy base 64 encoding of login and password>,Basic <AAF base 64 encoding of login and password>,YYYY-MM-DD

OOM deployments

1. AAF is enabled by default in OOM. To disable AAF for APPC in OOM, set a config value of enableAAF: false in an override file. See the example below:

```
appc:
  enabled: true
  config:
    enableAAF: false
```