Encrypt login and password for database and LDAP

Summary: this page describes how to encrypt logins and passwords for the CAST Dashboards/RestAPI:

1. when connecting to CAST Storage Service/PostgreSQL (for connections to the Measurement/Dashboard schemas and to the user role /authorization database)
2. when configuring LDAP authentication

Introduction

When configuring CAST Dashboard / RestAPI connections to **CAST Storage Service/PostgreSQL** (i.e. Measurement/Dashboard schemas, or user roles /authorizations database) or to an **LDAP server for corporate login mode**, logins and passwords are defined in the relevant configuration files in **clear text**. This therefore represents a potential security risk. If your organization requires these logins and passwords to be **encrypted**, you can use the following instructions to do so.

Note that this document already assumes that you have a working connection to your deployed CAST Dashboard or RestAPI.

Encrypting access to CAST Storage Service/PostgreSQL

For CAST Dashboards 1.x, encrypted CAST Storage Service/PostgreSQL credentials are only supported for Dashboards deployed on **Apache Tomcat 8 or above**.

To encrypt the login and password that are defined when configuring access to the CAST Storage Service/PostgreSQL instance where your Measurement /Dashboard schemas are located and to the CAST Storage Service/PostgreSQL instance where the User role/authorizations database is stored (by default this is called **cast_dashboards**), browse to the following **URL** to access the built in **login/password key generation** page:

```
http://<server>:[<port>]/<dashboard>/static/key.html
```

Login with a user (whether static list or Active Directory) that has the **ADMIN** role - by default no users have this role in either static list mode or in Active Directory mode - see **User authentication** for more information.

When successfully authenticated, you now need to enter the **credentials (login and password)** for each of your target CAST Storage Service /PostgreSQL instances (that you would ordinarily configure in the **context.xml / application.properties file** for accessing the Measurement/Dashboard Service and for the User role/authorizations database) and that you wish to encrypt. In the example below, we have entered the default credentials for a CAST Storage Service/PostgreSQL instance (**operator/CastAIP**):

Now click the **Encrypt** button - CAST will then generate a key that relates to the credentials you entered:

You now need to copy this key to the clipboard or to a text file and then follow the instructions below for your specific dashboard release.

For CAST Dashboards 1.x

Open the following file with a text editor:

```
CATALINA_HOME\webapps\<dashboard>\META-INF\context.xml
```

Find the following section of code and replace the line containing "**username**" and "**password**" with the **key** you generated previously:

```
<Resource name="jdbc/domains/AAD" url="jdbc:postgresql://localhost:2280/postgres"
    initConnectionSqls="SET search_path TO CAST_MEASURE;"
    username="operator" password="CastAIP"

    auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    validationQuery="select 1"
    initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>
```

For example:

```
<Resource name="jdbc/domains/AAD" url="jdbc:postgresql://localhost:2280/postgres"
    initConnectionSqls="SET search_path TO CAST_MEASURE;"
        key="D228ED8B5E5690B3A75"

    auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    validationQuery="select 1"
    initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>
```

Now add a new line directly underneath the line containing the "**key**" entry as follows - take note of the line that is specific to your release of CAST AIP and Apache Tomcat:

```
WARs delivered in CAST AIP  8.3.4 and all standalone CAST Dashboard Packages:

Tomcat  8 only: factory="com.castsoftware.adg.webservice.security.BasicDataSourceFactory"

WARs delivered in CAST AIP 8.3.0 - 8.3.3:

Tomcat 7: factory="com.castsoftware.adg.webservice.security.BasicDataSourceFactory"
Tomcat 8/8.5/9: factory="com.castsoftware.adg.webservice.security.BasicDataSourceFactory2"
```

Your database access resource should now look like this (this is an example for Tomcat 8 in CAST AIP 8.3.4 and all standalone CAST Dashboard Packages):

```
<Resource name="jdbc/domains/AAD" url="jdbc:postgresql://localhost:2280/postgres"
    initConnectionSqls="SET search_path TO CAST_MEASURE;"
    key="D228ED8B5E5690B3A75"
    factory="com.castsoftware.adg.webservice.security.BasicDataSourceFactory"

        auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    validationQuery="select 1"
    initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>
```

Save the file, **reload the cache** (see **Reload the cache**) and then reload your CAST Dashboard / RestAPI and ensure you can login and view the data you need to.

> You may need to repeat the above for each target CAST Storage Server/PostgreSQL instance resource you have configured in the **context.xml** file.

For CAST Dashboards 2.x

Open the following file with a text editor:

```
WAR  2.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\classes\application.properties

ZIP  2.x
<unpacked_zip>\application.properties
```

Find the following sections of code (one is for the application schema CSS/PostgreSQL host and the other is for the user management schema host) and replace the lines "**restapi.datasource[0].username**" / "**spring.datasource.username**" and "**restapi.datasource[0].password**" / "**spring.datasource.password**" with one single line containing your generated "**key**":

```
## DATASOURCE
# Resource1 is the datasource name used in domains.properties
# Adapt server name (localhost) and port (2282) if required
# You can add multiple datasources if you want to connect to multiple CSS Servers. Datasource name must be
unique
# You have to configure your domains names and relative schema names in domains.properties
restapi.datasource[0].url=jdbc:postgresql://localhost:2282/postgres
restapi.datasource[0].username=operator
restapi.datasource[0].password=CastAIP
restapi.datasource[0].poolname=Resource1
restapi.datasource[0].minimumIdle=10
restapi.datasource[0].maximumPoolSize=20

#datasource configuration for user management
spring.datasource.url=jdbc:postgresql://localhost:2282/postgres?ApplicationName=DASHBOARDS
¤tSchema=cast_dashboards
spring.datasource.platform=postgres
spring.datasource.username=operator
spring.datasource.password=CastAIP
spring.datasource.initialization-mode=always
spring.datasource.driver-class-name=org.postgresql.Driver
spring.liquibase.change-log=classpath:db/changelog/db.changelog-master.xml
spring.liquibase.default-schema=cast_dashboards
spring.liquibase.enabled=true
```

For example:

```
## DATASOURCE
# Resource1 is the datasource name used in domains.properties
# Adapt server name (localhost) and port (2282) if required
# You can add multiple datasources if you want to connect to multiple CSS Servers. Datasource name must be
unique
# You have to configure your domains names and relative schema names in domains.properties
restapi.datasource[0].url=jdbc:postgresql://localhost:2282/postgres
restapi.datasource[0].key=D228ED8B5E5690B3A75
restapi.datasource[0].poolname=Resource1
restapi.datasource[0].minimumIdle=10
restapi.datasource[0].maximumPoolSize=20

#datasource configuration for user management
spring.datasource.url=jdbc:postgresql://localhost:2282/postgres?ApplicationName=DASHBOARDS
¤tSchema=cast_dashboards
spring.datasource.platform=postgres
spring.datasource.key=D228ED8B5E5690B3A75
spring.datasource.initialization-mode=always
spring.datasource.driver-class-name=org.postgresql.Driver
spring.liquibase.change-log=classpath:db/changelog/db.changelog-master.xml
spring.liquibase.default-schema=cast_dashboards
spring.liquibase.enabled=true
```

Save the file, **reload the cache** (see **Reload the cache**) and then reload your CAST Dashboard / RestAPI and ensure you can login and view the data you need to.

> You may need to repeat the above for each target CAST Storage Server/PostgreSQL instance resource you have configured in the **application.properties** file.

Encrypting access to an LDAP server

When configuring access to an LDAP server for authentication, an LDAP **service account login** and **password** must be specified in the **.properties** file in clear text as described in **User authentication**:

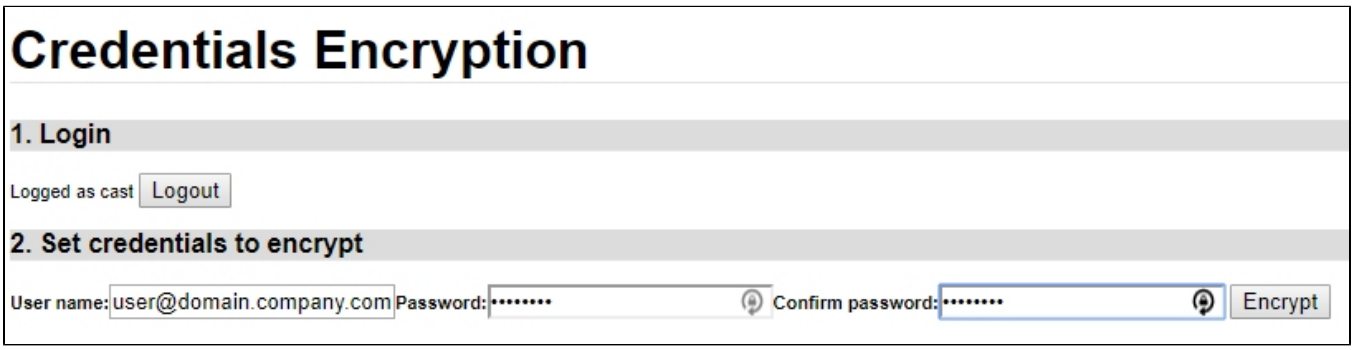
```
WAR 1.x
security.ldap.account.dn=cn=serviceaccount,dc=example,dc=com
security.ldap.account.password=password

WAR and ZIP  2.x
security.ldap.manager.dn=CN=serviceaccount,OU=RESOURCES,OU=FR,DC=example,DC=com
security.ldap.manager.password=password
```

To avoid the need to do this, browse to the following **URL** to access the built in **login/password key generation** page:

```
http://<server>:[<port>]/<dashboard>/static/key.html
```

Login with a user (whether Default Authentication or LDAP) that has the **ADMIN** role - by default no users have this role in either mode - see **User authentication** for more information:

When successfully authenticated, you now need to enter the **credentials (service account login and password)** for your LDAP server that you would ordinarily enter into the **.properties** file for configuring LDAP mode, and that you wish to encrypt. In the example below, we have entered the required LDAP credentials:



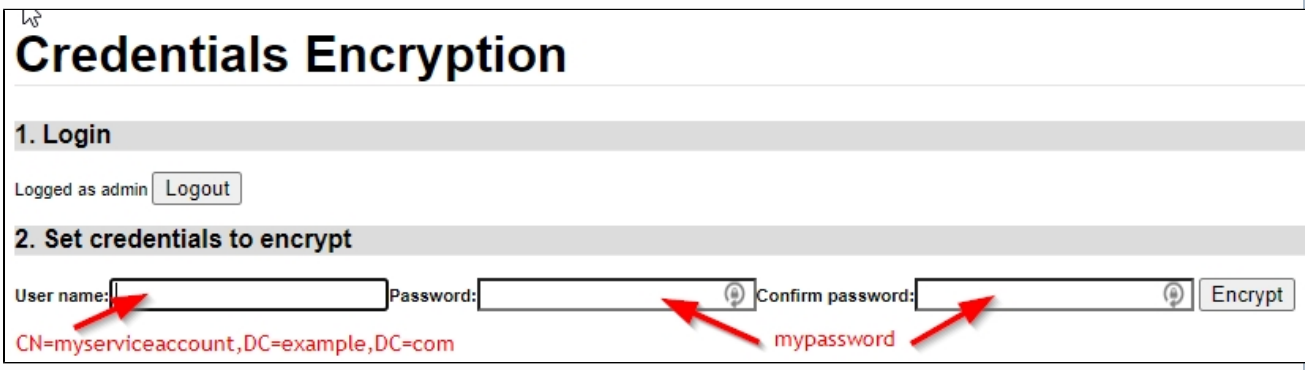
Note that the encryption key combines the values assigned to the following lines in the **.properties** file:

```
WAR 1.x
security.ldap.account.dn=cn=serviceaccount,dc=example,dc=com
security.ldap.account.password=password

WAR and ZIP  2.x
security.ldap.manager.dn=CN=serviceaccount,OU=RESOURCES,OU=FR,DC=example,DC=com
security.ldap.manager.password=password
```

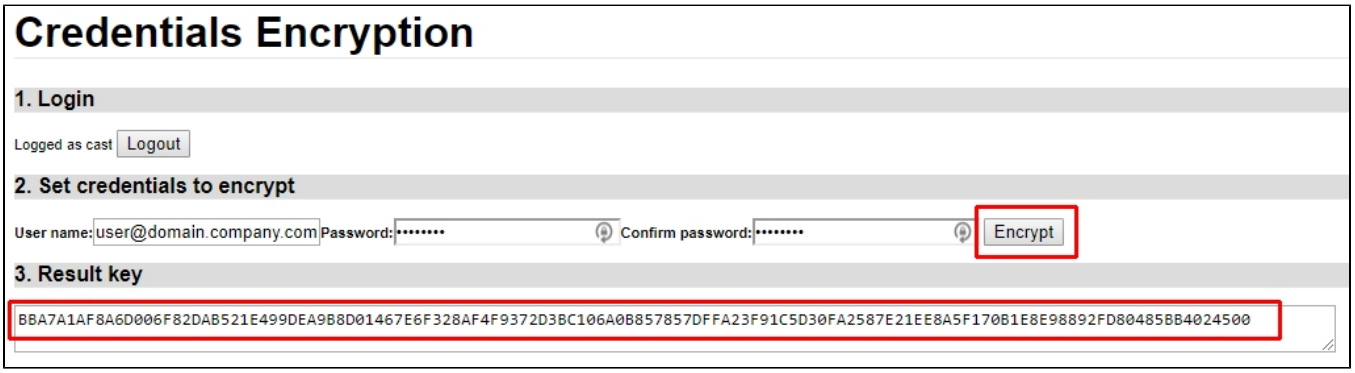
Therefore, you must enter in the "**username**" and "**password**" fields in the encryption tool EXACTLY what is entered in the "**dn=**" and "**password=**" lines in the **.properties** file. For example, if the **.properties** file contains:

```
WAR 1.x
security.ldap.account.dn=CN=myserviceaccount,DC=example,DC=com
security.ldap.account.password=mypassword

WAR and ZIP  2.x
security.ldap.manager.dn=CN=myserviceaccount,DC=example,DC=com
security.ldap.manager.password=mypassword
```

...then you need to enter exactly the same in the following fields:



Now click the **Encrypt** button - CAST will then generate a key that relates to the credentials you entered:



You now need to copy this key to the clipboard or to a text file and then open the following file with a text editor:

```
WAR 1.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\security.properties

WAR  2.x
CATALINA_HOME\webapps\<dashboard>\WEB-INF\classes\application.properties

ZIP  2.x
<unpacked_zip>\application.properties
```

Locate the following configuration in the file:

```
WAR 1.x
# Parameters for ldap mode
# ------------------------
security.ldap.url=ldap://directory.example.com/
security.ldap.account.dn=cn=serviceaccount,dc=example,dc=com
security.ldap.account.password=password
security.ldap.account.key=
security.ldap.usersearch.base=dc=example,dc=com
security.ldap.usersearch.filter=(&(objectClass=user)(sAMAccountName={0}))
security.ldap.groupsearch.base=dc=example,dc=com
security.ldap.groupsearch.filter=(&(objectClass=group)(member={0}))

WAR and ZIP  2.x

## SPRING SECURITY LDAP CONFIG
# LDAP url, in the form ldap://HOST:PORT
security.ldap.url=ldap://directory.example.com/
# The ldap base where users and groups can be found
security.ldap.base=dc=example,dc=com
# The DN for accessing the LDAP repository
security.ldap.manager.dn=CN=serviceaccount,OU=RESOURCES,OU=FR,DC=example,DC=com
# The associated password. You can encrypt this using the aip encryption tool
security.ldap.manager.password=password
```

For CAST Dashboards 1.x

First remove the two lines with the `security.ldap.account.dn` and `security.ldap.account.password` parameters. Then enter the key generated previously into the line containing "**key**". This should give you the following:

```
# Parameters for ldap mode
# ------------------------
security.ldap.url=ldap://directory.example.com/
security.ldap.account.key=A9762B77F8A5B6C0A885BABD58DFA1438D77A51B94ECA09
security.ldap.usersearch.base=dc=example,dc=com
security.ldap.usersearch.filter=(&(objectClass=user)(sAMAccountName={0}))
security.ldap.groupsearch.base=dc=example,dc=com
security.ldap.groupsearch.filter=(&(objectClass=group)(member={0}))
```

Save the file, **restart the web application** and ensure you can login and view the data you need to.

For CAST Dashboards 2.x

Add a new line underneath `security.ldap.manager.password` called `security.ldap.manager.key` and enter the key generated previous into this new line. In a development deployment you do not need to remove the `security.ldap.manager.dn` or `security.ldap.manager.password entries` - if the `security.ldap.manager.key` is present it will be used. **However, you SHOULD remove both lines in a live production environment so that the DN and password are not present in clear text:**

```
## SPRING SECURITY LDAP CONFIG
# LDAP url, in the form ldap://HOST:PORT
security.ldap.url=ldap://directory.example.com/
# The ldap base where users and groups can be found
security.ldap.base=dc=example,dc=com
# The DN for accessing the LDAP repository
security.ldap.manager.dn=CN=serviceaccount,OU=RESOURCES,OU=FR,DC=example,DC=com
# The associated password. You can encrypt this using the aip encryption tool
security.ldap.manager.password=password
security.ldap.manager.key=A9762B77F8A5B6C0A885BABD58DFA1438D77A51B94ECA09
```

Save the file, **restart the web application** and ensure you can login and view the data you need to.

What happens if the LDAP credentials change (new password)?

If your LDAP credentials change, for example a new password is generated on the LDAP server, then access to the the CAST Dashboard for any LDAP user will fail. As such the encryption key for the new credentials will need to be regenerated in the **key.html** page, however, this page requires authentication therefore it will not be accessible in order to generate a new key. This can only be resolved by:

- temporarily restoring access using a **login** and **password**, i.e. removing the `security.ldap.account.key / security.ldap.manager.key` line from the **.properties** file and (for 1.x WAR files only) re-adding the `security.ldap.account.dn` and `security.ldap.account.password` lines.
- accessing **key.html** and encrypting the new login/password into a key.
- re-adding the `security.ldap.account.key / security.ldap.manager.key` line with the new key and (for 1.x WAR files only) removing the `security.ldap.account.dn` and `security.ldap.account.password` lines.