Authentication and Authorization (SSO) Guide

This document provides information on how authentication works with SSO for GSRS system.

Typical Authentication Framework  
GSRS has a default built-in username/password authentication system. It is mostly used during development. For production systems, most organizations have their own authentication services they want to use. The default GSRS default authentication can be set to be bypassed in production environments. The picture below shows a typical authentication framework to use SSO for authentication.

Diagram

Description automatically generated

Figure 1: Authentication Structure

## Settings to Bypass GSRS Username/Password Authentication

To bypass the default GSRS password authentication, some configs need to be set in the GSRS config file.   
These settings instruct GSRS to trust headers coming from the proxy and set the header names that will contain the username and email for a given user.  
*## START AUTHENTICATION  
# SSO HTTP proxy authentication settings  
ix.authentication.trustheader=true  
ix.authentication.usernameheader = ${?AUTH\_USERNAME}  
ix.authentication.useremailheader = ${?AUTH\_EMAIL}  
# set this "false" to only allow authenticated users to see the application  
ix.authentication.allownonauthenticated=false  
# set this "true" to allow any user that authenticates to be registered as a user automatically  
ix.authentication.autoregister=false  
#Set this to "true" to allow autoregistered users to be active as well  
ix.authentication.autoregisteractive=false  
## END AUTHENTICATION  
ix.authentication.trustheader=true  
ix.authentication.autoregister=true*If the above are true, then the user with the username and email are passed to GSRS. If the user does not exist, then it will be created and assigned a password and key. By hitting the whoami endpoint the person can get their key to interact with the API.

## An Example SSO Setup in Deployment Environment

It uses Oracle server as the reverse proxy.

First, log on as root

----------------------------------------------------------------

edit the file

/etc/security/limits.conf

add these lines (including asterisks) to the end of it

\* soft nofile 4096

\* hard nofile 65536

\* soft nproc 2047

\* hard nproc 16384

----------------------------------------------------------------

The remainder of these instructions is inspired (though not copied verbatim) from:

https://oracle-base.com/articles/12c/oracle-http-server-ohs-installation-on-oracle-linux-6-and-7-1221

----------------------------------------------------------------

yum install binutils -y

yum install gcc -y

yum install gcc-c++ -y

yum install glibc -y

yum install glibc.i686 -y

yum install glibc-devel -y

yum install libaio -y

yum install libaio-devel -y

yum install libgcc -y

yum install libgcc.i686 -y

yum install libstdc++ -y

yum install libstdc++.i686 -y

yum install libstdc++-devel -y

yum install ksh -y

yum install make -y

yum install sysstat -y

yum install numactl -y

yum install numactl-devel -y

yum install motif -y

yum install motif-devel -y

----------------------------------------------------------------

groupadd -g 54321 oinstall

useradd -u 54321 -g oinstall orafda

usermod -a -G oinstall orafda

passwd XXXXXX

----------------------------------------------------------------

chmod a+rw /u01

----------------------------------------------------------------

# IMPORTANT

#

# IMPORTANT

#

# Now log on as orafda

# IMPORTANT

#

su orafda

----------------------------------------------------------------

mkdir -p /u01/app/oracle/product/12.2.1.4/ohs\_1

mkdir -p /u01/app/oracle/product/8/jdk\_1

mkdir -p /u01/app/oracle/config/domains

mkdir -p /u01/app/oracle/config/applications

chown -R orafda:oinstall /u01/app/oracle

chmod -R 775 /u01/app/oracle

mkdir -p /u01/app/software/

chown -R orafda:oinstall /u01/app/software/

chmod -R 775 /u01/app/software/

mkdir -p ~orafda/scripts

# this last one may already exist -- that's OK

----------------------------------------------------------------

cat > ~orafda/scripts/status.sh <<EOF

echo .

echo .

echo .

ps -ef | grep httpd | head -n1

echo .

echo .

echo .

ps -p $(cat /u01/app/oracle/config/domains/ohs1/system\_components/OHS/ohs1/data/nodemanager/ohs1.lck) &> /dev/null && echo up || echo down

echo .

echo .

echo .

cat /u01/app/oracle/config/domains/ohs1/system\_components/OHS/ohs1/data/nodemanager/ohs1.state

echo .

echo .

echo .

EOF

----------------------------------------------------------------

cat > ~orafda/scripts/setEnv.sh <<EOF

export ORACLE\_BASE=/u01/app/oracle

export MW\_HOME=\$ORACLE\_BASE/product/12.2.1.4/fmw\_1

export JAVA\_HOME=\$ORACLE\_BASE/product/8/jdk\_1

export PATH=\$JAVA\_HOME/bin:\$PATH

export ORACLE\_HOME=\$MW\_HOME

export DOMAIN\_HOME=\$ORACLE\_BASE/config/domains/ohs1

export INSTANCE\_HOME=\$DOMAIN\_HOME/config/fmwconfig/components/OHS/instances/ohs1

EOF

----------------------------------------------------------------

echo ". ~orafda/scripts/setEnv.sh" >> ~orafda/.bash\_profile

. ~orafda/scripts/setEnv.sh

----------------------------------------------------------------

cat > ~orafda/scripts/start\_all.sh <<EOF

#!/bin/bash

. ~orafda/scripts/setEnv.sh

echo "Start Node Manager" `date`

nohup \$DOMAIN\_HOME/bin/startNodeManager.sh > /dev/null 2>&1 &

sleep 10

echo "Start OHS" `date`

\$DOMAIN\_HOME/bin/startComponent.sh ohs1

echo "Complete..." `date`

EOF

----------------------------------------------------------------

cat > ~orafda/scripts/stop\_all.sh <<EOF

#!/bin/bash

. ~orafda/scripts/setEnv.sh

echo "Stop OHS..." `date`

\$DOMAIN\_HOME/bin/stopComponent.sh ohs1

echo "Stop Node Manager..." `date`

\$DOMAIN\_HOME/bin/stopNodeManager.sh

echo "Complete..." `date`

EOF

----------------------------------------------------------------

chmod a+rx ~orafda/scripts/\*.sh

----------------------------------------------------------------

#Create Service File "/etc/systemd/system/ohs.service"

------------------

[Unit]

Description=ohs process

After=network.target centrifydc.service sshd.service

[Service]

User=orafda

Type=forking

ExecStart=/bin/sh /var/opt/oracle/scripts/start\_all.sh start

ExecStop=/bin/sh /var/opt/oracle/scripts/stop\_all.sh stop

[Install]

WantedBy=multi-user.target

Alias=ohs.service

------------------

chmod 664 /etc/systemd/system/ohs.service

systemctl daemon-reload

systemctl enable ohs

systemctl is-enabled ohs.service

----------------------------------------------------------------

#Create Response File "/u01/app/software/webtier.rsp"

#It is used to install a Standalone HTTP Server and contains a place-holder for the ORACLE\_HOME, which we will set as part of the installation.

------------------

[ENGINE]

#DO NOT CHANGE THIS.

Response File Version=1.0.0.0.0

[GENERIC]

#Set this to true if you wish to skip software updates

DECLINE\_AUTO\_UPDATES=true

#My Oracle Support User Name

MOS\_USERNAME=

#My Oracle Support Password

MOS\_PASSWORD=<SECURE VALUE>

#If the Software updates are already downloaded and available on your local system, then specify the path to the directory where these patches are available and set SPECIFY\_DOWNLOAD\_LOCATION to true

AUTO\_UPDATES\_LOCATION=

#Proxy Server Name to connect to My Oracle Support

SOFTWARE\_UPDATES\_PROXY\_SERVER=

#Proxy Server Port

SOFTWARE\_UPDATES\_PROXY\_PORT=

#Proxy Server Username

SOFTWARE\_UPDATES\_PROXY\_USER=

#Proxy Server Password

SOFTWARE\_UPDATES\_PROXY\_PASSWORD=<SECURE VALUE>

#The oracle home location. This can be an existing Oracle Home or a new Oracle Home

ORACLE\_HOME=###ORACLE\_HOME###

#Set this variable value to the Installation Type selected as either Standalone HTTP Server (Managed independently of WebLogic server) OR Collocated HTTP Server (Managed through WebLogic server)

INSTALL\_TYPE=Standalone HTTP Server (Managed independently of WebLogic server)

#Provide the My Oracle Support Username. If you wish to ignore Oracle Configuration Manager configuration provide empty string for user name.

MYORACLESUPPORT\_USERNAME=

#Provide the My Oracle Support Password

MYORACLESUPPORT\_PASSWORD=<SECURE VALUE>

#Set this to true if you wish to decline the security updates. Setting this to true and providing empty string for My Oracle Support username will ignore the Oracle Configuration Manager configuration

DECLINE\_SECURITY\_UPDATES=true

#Set this to true if My Oracle Support Password is specified

SECURITY\_UPDATES\_VIA\_MYORACLESUPPORT=false

#Provide the Proxy Host

PROXY\_HOST=

#Provide the Proxy Port

PROXY\_PORT=

#Provide the Proxy Username

PROXY\_USER=

#Provide the Proxy Password

PROXY\_PWD=<SECURE VALUE>

#Type String (URL format) Indicates the OCM Repeater URL which should be of the format [scheme[Http/Https]]://[repeater host]:[repeater port]

COLLECTOR\_SUPPORTHUB\_URL=

------------------

chmod a+rw /u01/app/software/webtier.rsp

------------------------------------------------------------------------

#Specify an Oracle inventory location.

#Create a file called "/u01/app/software/oraInst.loc" with the following contents.

------------------

inventory\_loc=/u01/app/oraInventory

inst\_group=oinstall

------------------

chmod a+rw /u01/app/software/oraInst.loc

------------------------------------------------------------------------

# IMPORTANT

#

# IMPORTANT

#

# exit orafda and return to root

------------------------------------------------------------------------

#initiate the installation in silent mode.

# Unzip the software.

cd /u01/app/software

unzip -o fmw\_12.2.1.4.0\_ohs\_linux64\_Disk1\_1of1.zip

source ~orafda/scripts/setEnv.sh

# Set the ORACLE\_HOME in the response file.

sed -i -e "s|###ORACLE\_HOME###|$ORACLE\_HOME|g" /u01/app/software/webtier.rsp

mkdir /u01/app/software/temp

chmod a+x /u01/app/software/temp

# Install OHS.

./fmw\_12.2.1.4.0\_ohs\_linux64.bin -J-Djava.io.tmpdir=/u01/app/software/temp -ignoreSysPrereqs -silent -responseFile /u01/app/software/webtier.rsp -invPtrLoc /u01/app/software/oraInst.loc

------------------------------------------------------------------------

# Create OHS Instance

cd $MW\_HOME/oracle\_common/common/bin

nano ~orafda/scripts/create\_ohs.py

----------------

selectTemplate('Oracle HTTP Server (Standalone)')

loadTemplates()

domain\_path = raw\_input('Enter domain path: ').strip()

nmpass = raw\_input('Enter nodemanager password: ').strip()

cd('/SecurityConfiguration/base\_domain')

set('NodeManagerPasswordEncrypted',nmpass)

writeDomain(domain\_path)

exit()

----------------

------------------------------------------------------------------------

/u01/app/oracle/product/12.2.1.4/fmw\_1/oracle\_common/common/bin/wlst.sh ~orafda/scripts/create\_ohs.py

--> Enter domain path: /u01/app/oracle/config/domains/ohs1

--> Enter nodemanager password: XXXXXX (the password must be at least 8 characters and contain at least one numeral, otherwise you'll see a very unhelpful error)

# the log for this particular script run can be found at /tmp/wlstOfflineLogs\_orafda/cfgfwk\_<date\_time>.log

# Verify successful creation of the "ohs1" instance:

ls -al /u01/app/oracle/config/domains/ohs1

------------------------------------------------------------------------

# First-time Start:

# Make sure the environment is set

#

source ~orafda/scripts/setEnv.sh

cd /u01/app/oracle/config/domains/ohs1

# Start the node manager

#

nohup $DOMAIN\_HOME/bin/startNodeManager.sh > /u01/app/oracle/config/domains/ohs1/nohup\_ohs\_$(date +"%Y\_%m\_%d\_\_%Hh\_%Mm\_%Ss\_%p").out 2>&1 & disown

# The first time we start the OHS instance, we use the "storeUserConfig" option which saves the credentials so we never need to enter them again.

# Enter the password (XXXXXX) when prompted

#

$DOMAIN\_HOME/bin/startComponent.sh ohs1 storeUserConfig

------------------------------------------------------------------------

# Deploy the webgate as per the instructions here:

# https://docs.oracle.com/en/middleware/fusion-middleware/12.2.1.4/wtins/configuring-webgate-oracle-access-manager.html#GUID-B4B416E5-A4F1-43E1-972B-4375221E5182

#

# on that page, DOMAIN\_HOME = /u01/app/oracle/config/domains/ohs1/config/fmwconfig/components/OHS/ohs1

cd $ORACLE\_HOME/webgate/ohs/tools/deployWebGate

./deployWebGateInstance.sh -w /u01/app/oracle/config/domains/ohs1/config/fmwconfig/components/OHS/ohs1 -oh $ORACLE\_HOME

ls -lart /u01/app/oracle/config/domains/ohs1/config/fmwconfig/components/OHS/ohs1/webgate/

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$ORACLE\_HOME/lib

cd /u01/app/oracle/product/12.2.1.4/fmw\_1/webgate/ohs/tools/setup/InstallTools/

./EditHttpConf -w /u01/app/oracle/config/domains/ohs1/config/fmwconfig/components/OHS/ohs1 -oh $ORACLE\_HOME

------------------------------------------------------------------------

IMPORTANT:

Create an incident ticket with OIMT DAS FDA SSO SUPPORT to exclude /api/v1/\*\* from the sso policy.

In the ticket, include the webgate host's name

------------------------------------------------------------------------

nano $DOMAIN\_HOME/config/fmwconfig/components/OHS/instances/ohs1/ohs.plugins.nodemanager.properties

# put this line at the end of it, save, and exit:

environment.LD\_LIBRARY\_PATH=$ORACLE\_HOME/webgate/ohs/lib/:$ORACLE\_HOME/ohs/lib:$ORACLE\_HOME/oracle\_common/lib:$ORACLE\_HOME/lib

~orafda/scripts/stop\_all.sh

~orafda/scripts/start\_all.sh

------------------------------------------------------------------------

cd ~orafda/scripts/

nano ohs\_status.py

---------------

nmConnect('', '', 'localhost', '5556', 'base\_domain', '/home/oracle/product/12g/12.1.3/ohs2/oh/user\_projects/domains/base\_domain','ssl')

nmServerStatus('ohs1','OHS')

---------------

cd ~orafda/scripts/

nano ohs\_status.sh

---------------

$ORACLE\_HOME/oracle\_common/common/bin/wlst.sh ~orafda/scripts/ohs\_status.py

---------------

chmod a+x ohs\_status.sh

./ohs\_status.sh

--> enter the password, XXXXXX

------------------------------------------------------------------------

get a file named something like "WG\_66020.zip" from the SSO team (or the Middleware team)

put it at /u01/app/oracle/config/domains/ohs1/config/fmwconfig/components/OHS/ohs1/webgate/config/

unzip it there

IMPORTANT: make sure you're still logged in as orafda

-----------------------------------------------------

whoami

-----------------------------------------------------

restart OHS -- but first, make sure you're still logged in as orafda

-----------------------------------------------------

~/scripts/stop\_all.sh

~/scripts/start\_all.sh

-----------------------------------------------------

Test the instance

-----------------------------------------------------

curl http://fdslv66020.fda.gov:7777/

-----------------------------------------------------

------------------------------------------------------------------------

Make sure that the file:

/u01/app/oracle/config/domains/ohs1/config/fmwconfig/components/OHS/ohs1/httpd.conf

contains the line:

IncludeOptional "moduleconf/\*.conf"

cd /u01/app/oracle/config/domains/ohs1/config/fmwconfig/components/OHS/ohs1/moduleconf

# create a file named virtual\_hosts.conf

#

nano virtual\_hosts.conf

----------------

#<LocationMatch "^/api.\*">

#AuthType Oblix

#require valid-user

#</LocationMatch>

<Location "/api">

RequestHeader unset OAM\_REMOTE\_USER

AuthType None

Require all granted

</Location>

<VirtualHost \*:7777>

ServerName https://gsrs-uat.preprod.fda.gov # <<<<<<<<< ------------------------------------- change the host name as needed

ProxyPreserveHost On

ProxyPassMatch ^/index.html !

ProxyPassMatch ^/$ !

ProxyPassMatch ^/ginas/app/index.html !

ProxyPassMatch ^/ginas/app$ !

ProxyPassMatch ^/ginas/app/$ !

ProxyPass / http://localhost:8080/

ProxyPassReverse / http://localhost:8080/

</VirtualHost>

----------------

------------------------------------------------------------------------

Restrict SSO auth to only non-/api routes

-----------------------------------------

cd /u01/app/oracle/config/domains/ohs1/config/fmwconfig/components/OHS/instances/ohs1/

nano webgate.conf

# comment out this section in the file and insert the replacement section as shown here:

----------------

#<LocationMatch "^/.\*">

#AuthType Oblix

#require valid-user

#</LocationMatch>

<LocationMatch "^(?!/api)/[^/]+">

AuthType Oblix

require valid-user

</LocationMatch>

----------------

------------------------------------------------------------------------

cd /u01/app/oracle/config/domains/ohs1/config/fmwconfig/components/OHS/ohs1/htdocs

mv index.html index.html.orig

nano index.html

----------------

<!DOCTYPE html>

<html>

<head>

<script>

window.location = "/ginas/app/beta/home";

</script>

</head>

<body>

Launching GSRS ...

<p>

<a href="/ginas/app">Click here if you don't see GSRS within 3 seconds</a>

</body>

</html>

----------------

mkdir ginas

mkdir ginas/app

cp index.html ginas/app

------------------------------------------------------------------------

# IMPORTANT

#

# IMPORTANT

#

# exit orafda and return to root

------------------------------------------------------------------------

shut off GSRS

rm 12.2.1.3 or 4 dir

rm ora\_inventory

edit .rsp file

install

create ohs instance

instructions to set up webgate

run 2 scripts that I receive from the SSO team

get zip file to configure webgate

create virtual host for GSRS with a proxy pass to GSRS on port 9000

------------------------------------------------------------------------