Configure ISE 2.2 for integration with MySQL server

Contents

Introduction

This document describes how to configure a Cisco Identity Services Engine (ISE) 2.2 for integration with MySQL Open Database Connectivity (ODBC) external source. This document is valid for setups that use MySQL as the external identity source for the ISE authentication and authorization.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Identity Services Engine (ISE) configuration
- Basic MySQL configuration

Components Used

The information this document is based on these software and hardware versions:

- Cisco ISE Version 2.2

- Ubuntu Linux with MySQL installed
- Cisco Wireless LAN Controller (WLC) Version 8.0.100.0
- Microsoft Windows Version 7x64

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background information

ISE 2.2 supports multiple ODBC external sources, one of them is MySQL. You can use ODBC as external identity source to authenticate users and endpoints similar to Active Directory (AD). ODBC identity source can be used in an identity store sequence and for Guest and Sponsor authentications.

This is a list database engines supported in ISE 2.2:

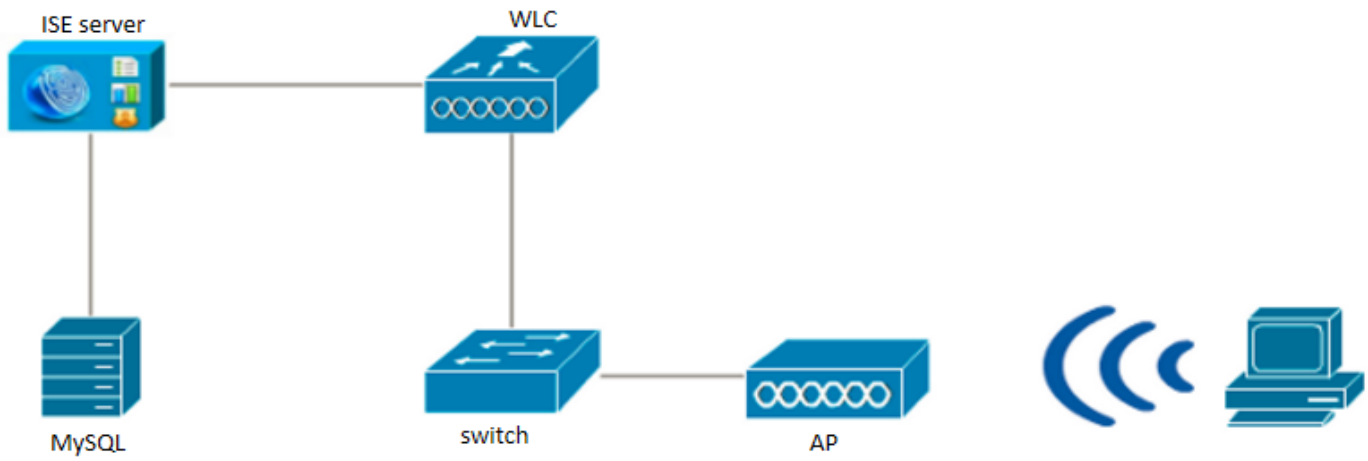
- MySQL
- Oracle
- PostgreSQL
- Microsoft SQL Server
- Sybase

More information can be found here: https://www.cisco.com/c/en/us/td/docs/security/ise/2-2/admin_guide/b_ise_admin_guide_22/b_ise_admin_guide_22_chapter_01101.html#concept_6EB9B4875CBB47D79168E329696E2C65

Configure

Network Diagram

In this configuration example, the endpoint uses a wireless adapter in order to associate with the wireless network. The Wireless LAN (WLAN) on the WLC is configured in order to authenticate the users via the ISE. On the ISE, MySQL is configured as an external identity store. This image illustrates the network topology that is used:



Configurations

MySQL configuration presented is an example. Do not treat is as a Cisco recomendation.

1. Configure MySQL on Ubuntu:

Update your system:

```
sudo apt-get update
sudo apt-get upgrade
```

Install MySQL (you should be prompted for a password for root user during the installation):

```
sudo apt-get install mysql-server
```

To access MySQL database:

```
mysql -u root -p
```

2. Configure database and tables:

Create database:

```
mysql>
mysql> CREATE DATABASE demo_db;
Query OK, 1 row affected (0.00 sec)
mysql>
mysql> use demo_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
```

Create database user and grant him privileges:

```
mysql>
mysql> CREATE USER 'cisco' IDENTIFIED BY 'cisco';
mysql> GRANT USAGE ON *.* TO 'cisco'@'%';
mysql> GRANT ALL PRIVILEGES ON `demo_db`.* TO 'cisco'@'%';
mysql> GRANT SELECT ON *.* TO 'cisco'@'%';
```

Create table of users:

```
mysql>
mysql> CREATE TABLE ´users´ (
-> `user_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
-> `username` varchar(50) NOT NULL,
-> `password` varchar(50) NOT NULL,
-> PRIMARY KEY (`user_id`),
-> UNIQUE KEY `username_UNIQUE` (`username`)
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.01 sec)
```

Create users and add them into the table:

```
mysql>
mysql> INSERT INTO users
-> (user_id, username, password)
-> VALUES
-> (1, "alice", "Krakow123");
Query OK, 1 row affected (0.00 sec)
```

You can add other users similarly and list the content of the table (the same way as users, add MAC address for MAB authentication - password can stay blank):

```
mysql>
mysql> select * from users;
```

```
+---------+----------+-----------+
| user_id | username | password  |
+---------+----------+-----------+
| 1       | alice    | Krakow123 |
| 2       | bob      | Krakow123 |
| 3       | oscar    | Krakow123 |
+---------+----------+-----------+
```

Create table of groups:

```
mysql>
mysql> CREATE TABLE `groups` (
-> `group_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
-> `groupname` varchar(50) NOT NULL,
-> PRIMARY KEY (`group_id`),
-> UNIQUE KEY `groupname_UNIQUE` (`groupname`)
-> ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.01 sec)
```

Create groups and add them into table:

```
mysql>
mysql> INSERT INTO groups
-> (group_id, groupname)
-> VALUES
-> (1, "everyone");
Query OK, 1 row affected (0.00 sec)
```

You can add other groups similarly and list the content of the table:

```
mysql>
mysql> select * from groups;
+----------+------------+
| group_id | groupname  |
+----------+------------+
| 3        | contractor |
| 2        | employee   |
| 1        | everyone   |
+----------+------------+
```

Create table for mappings between users and groups

```
mysql>
mysql> CREATE TABLE `user_group` (
-> `user_id` int(10) unsigned NOT NULL,
-> `group_id` int(10) unsigned NOT NULL,
-> PRIMARY KEY (`user_id`,`group_id`),
-> KEY `group_id` (`group_id`),
-> CONSTRAINT `user_group_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`)
-> ON DELETE CASCADE,
-> CONSTRAINT `user_group_ibfk_2` FOREIGN KEY (`group_id`) REFERENCES `groups`
-> (`group_id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
Query OK, 0 rows affected (0.01 sec)
```

Fill the table for mappings between users and groups

```
mysql>
mysql> INSERT INTO user_group
-> (user_id, group_id)
-> VALUES
```

```
-> (1, 1);
Query OK, 1 row affected (0.00 sec)
```

You can add other mappings similarly and list the content of the table:

```
mysql>
mysql> select * from user_group;
+---------+----------+
| user_id | group_id |
+---------+----------+
| 1       | 1        |
| 2       | 1        |
| 1       | 2        |
| 2       | 3        |
+---------+----------+
4 rows in set (0.00 sec)
```

3. Configure stored procedures

You have to configure the required stored procedures to authenticate users against an ODBC identity source. The tasks that are performed by procedure vary, based on the authentication protocol. ISE supports three different types of credential check against ODBC external store. You need to configure separate stored procedure for each type of check. ISE calls the appropriate stored procedure with input parameters and receives the output. The database can return a recordset or a set of named parameters in response to an ODBC query.

- **Plain text password authentication in ODBC database -** Authentication for PAP and PEAP occurs within the database. If procedure finds a username/password combination that matches the input, the user is successfully authenticated.
- **Plain text password fetching from ODBC database -** Authentication for CHAP, MS-CHAPv1/v2, EAP-MD5, LEAP, and EAP-MSCHAPv2 (as inner method of PEAP or EAP-FAST) occurs within Cisco ISE (ISE checks password provided by user and compares it with password received from stored procedure). The stored procedure returns the password if the username is correct. If the username is not found, it returns an error code.
- **Lookup** - Authentication for MAB occurs within the database. If the required username is found, relevant parameters are returned to ISE.

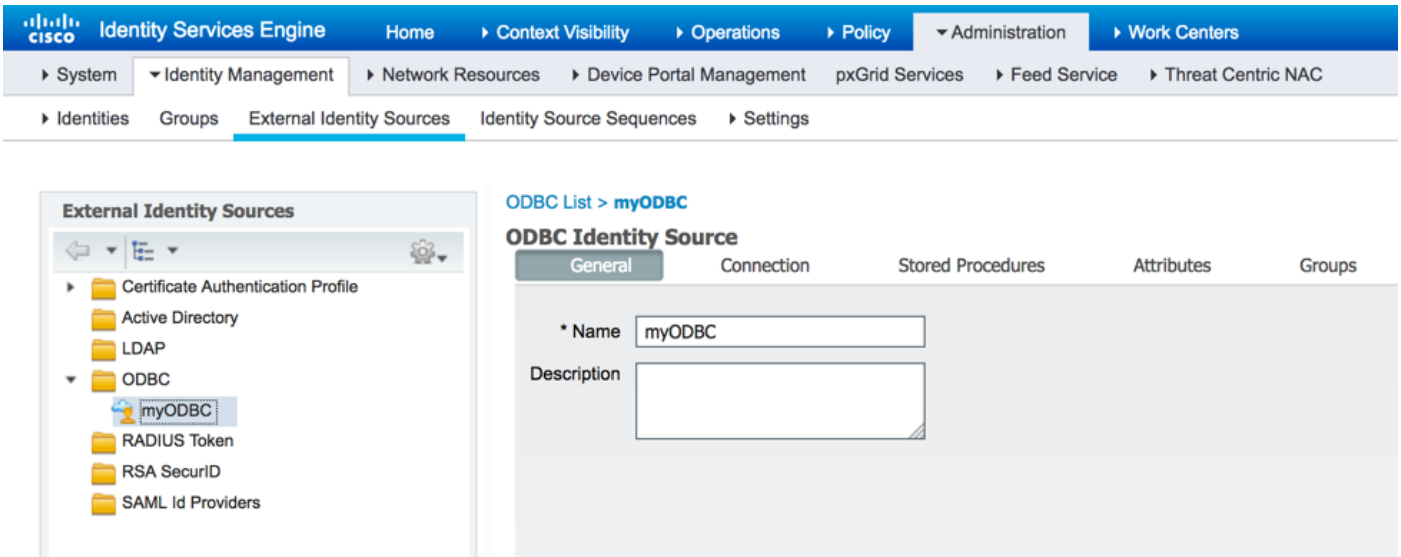
Each of those procedures should be defined with delimiter for MySQL to accept the syntax of the query:

```
DELIMITER //
CREATE DEFINER=`root`@`localhost` PROCEDURE `ISEGroups`(username varchar(64), OUT result INT)
begin
CASE username
WHEN '*' THEN
select distinct groupname from groups;
ELSE
select groupname from user_group
inner join users ON users.user_id = user_group.user_id
inner join groups ON groups.group_id = user_group.group_id
where users.username = username;
END CASE;
SET result = 0;
end //DELIMITER //
CREATE DEFINER=`root`@`localhost` PROCEDURE `ISEAuthUserPlainReturnsRecordset`(username
varchar(64), password varchar(255))
begin
IF EXISTS (select * from users where users.username = username and users.password = password )
THEN
```

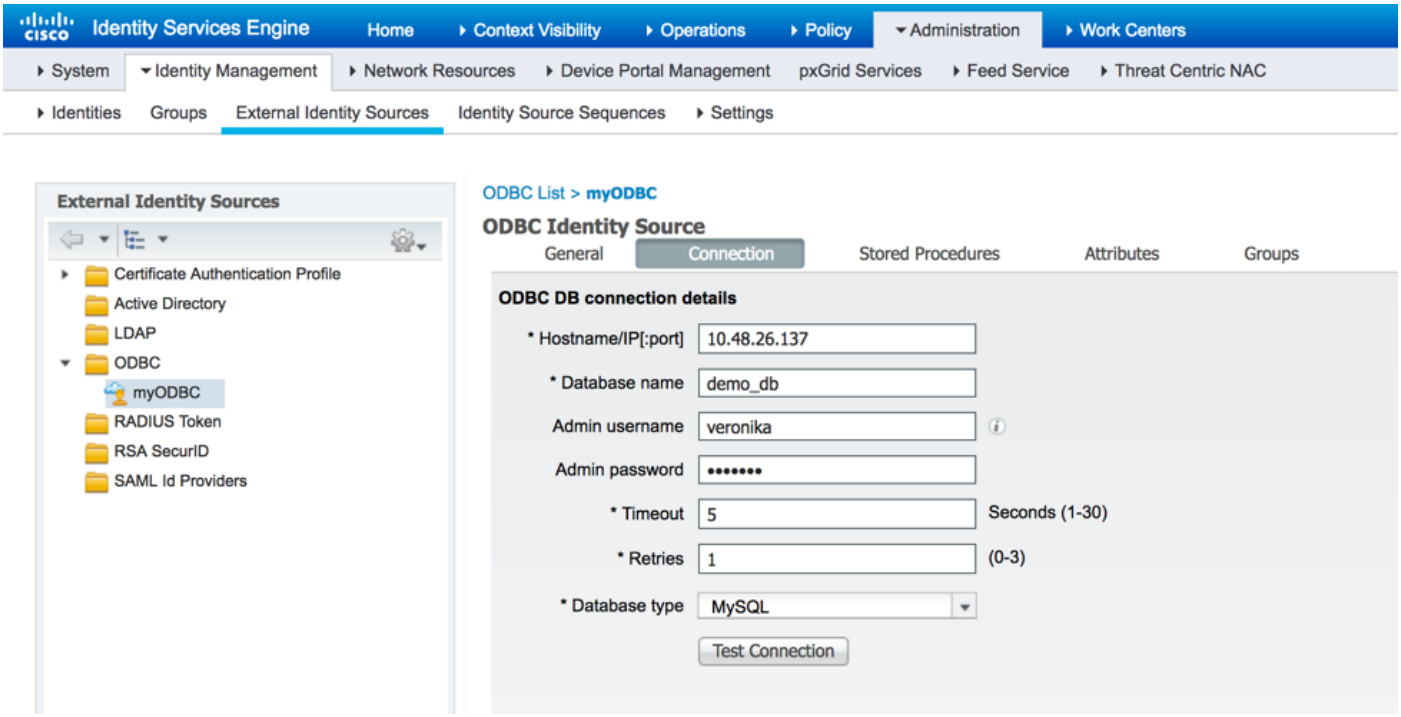
```
select 0,11,'This is a very good user, give him all access','no error';
ELSE
select 3, 0, 'odbc','ODBC Authen Error';
END IF;
end //DELIMITER //
CREATE DEFINER=`root`@`localhost` PROCEDURE `ISEFetchPasswordReturnsRecordset`(username
varchar(64))
begin
IF EXISTS (select * from users where users.username = username) THEN
select 0,11,'This is a very good user, give him all access','no error',password from users where
users.username = username;
ELSE
select 3, 0, 'odbc','ODBC Authen Error';
END IF;
end //DELIMITER //
CREATE DEFINER=`root`@`localhost` PROCEDURE `ISEUserLookupReturnsRecordset`(username
varchar(64))
begin
IF EXISTS (select * from users where users.username = username) THEN
select 0,11,'This is a very good user, give him all access','no error';
ELSE
select 3, 0, 'odbc','ODBC Authen Error';
END IF;
end //
```

4. Integrate ISE with MySQL:

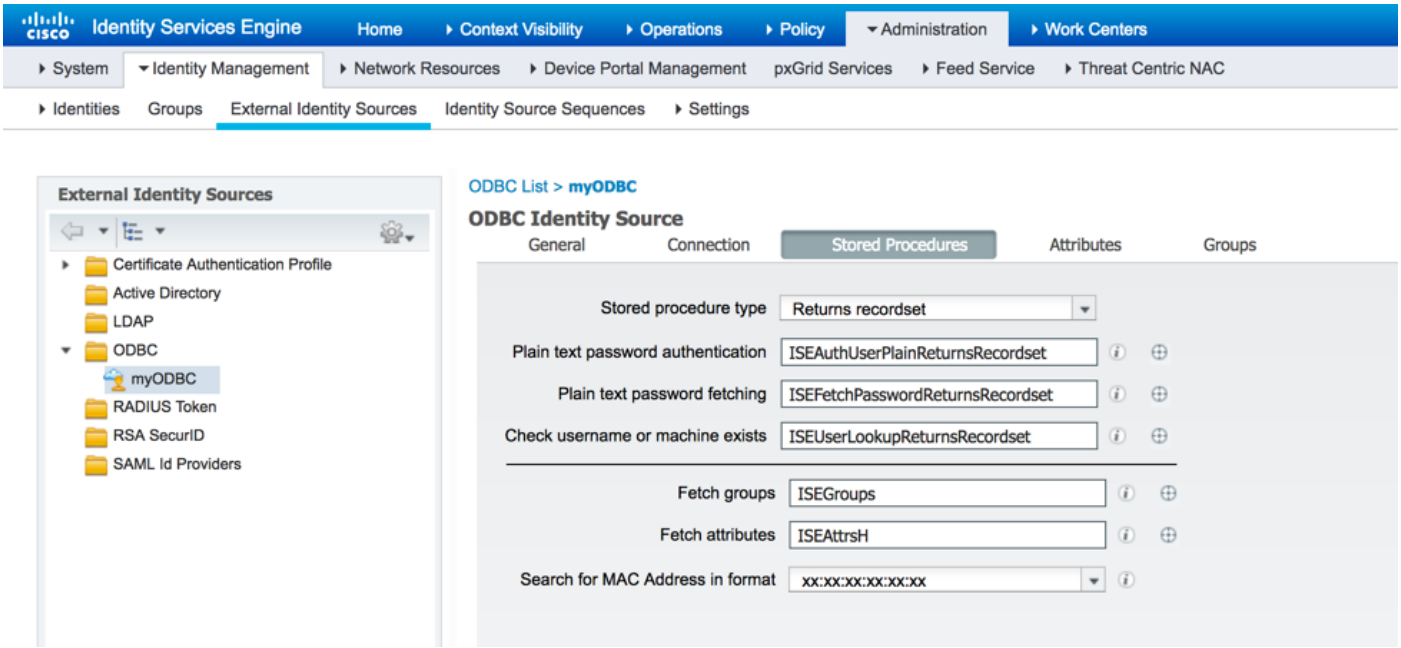
Use the information mentioned below in order to integrate MySQL with Cisco ISE. Navigate to **Administration > Identity Management > External Identity Sources > ODBC** and add new store:



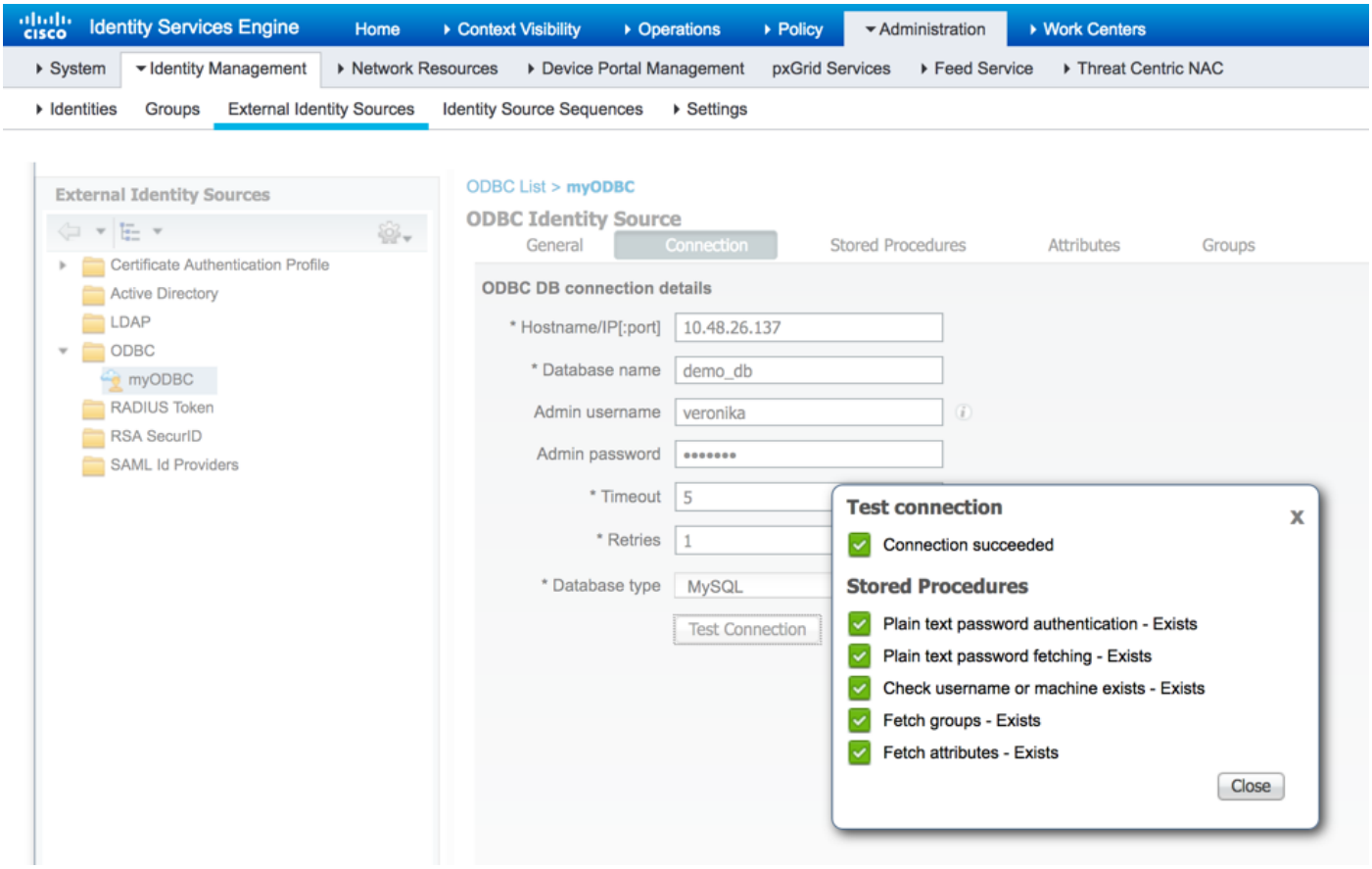
Use the IP address of Ubuntu that is running MySQL database as a hostname/IP address below. Specify type of database (in this situation MySQL is used), insert also database name and database user credentials that were created earlier:



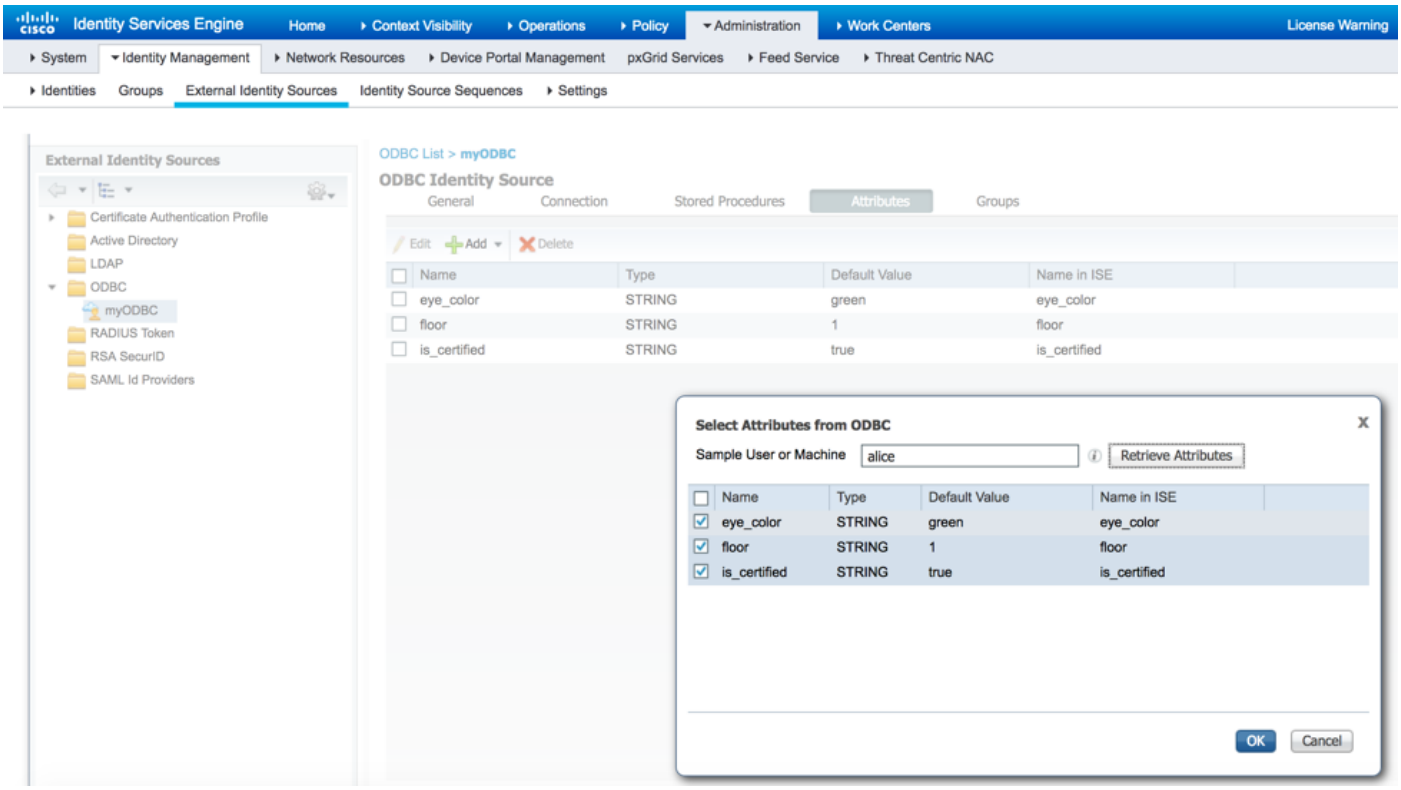
Specify the names of procedures that were created in MySQL – you need to be carefull with MAC address format (in this example it was changed to different format):



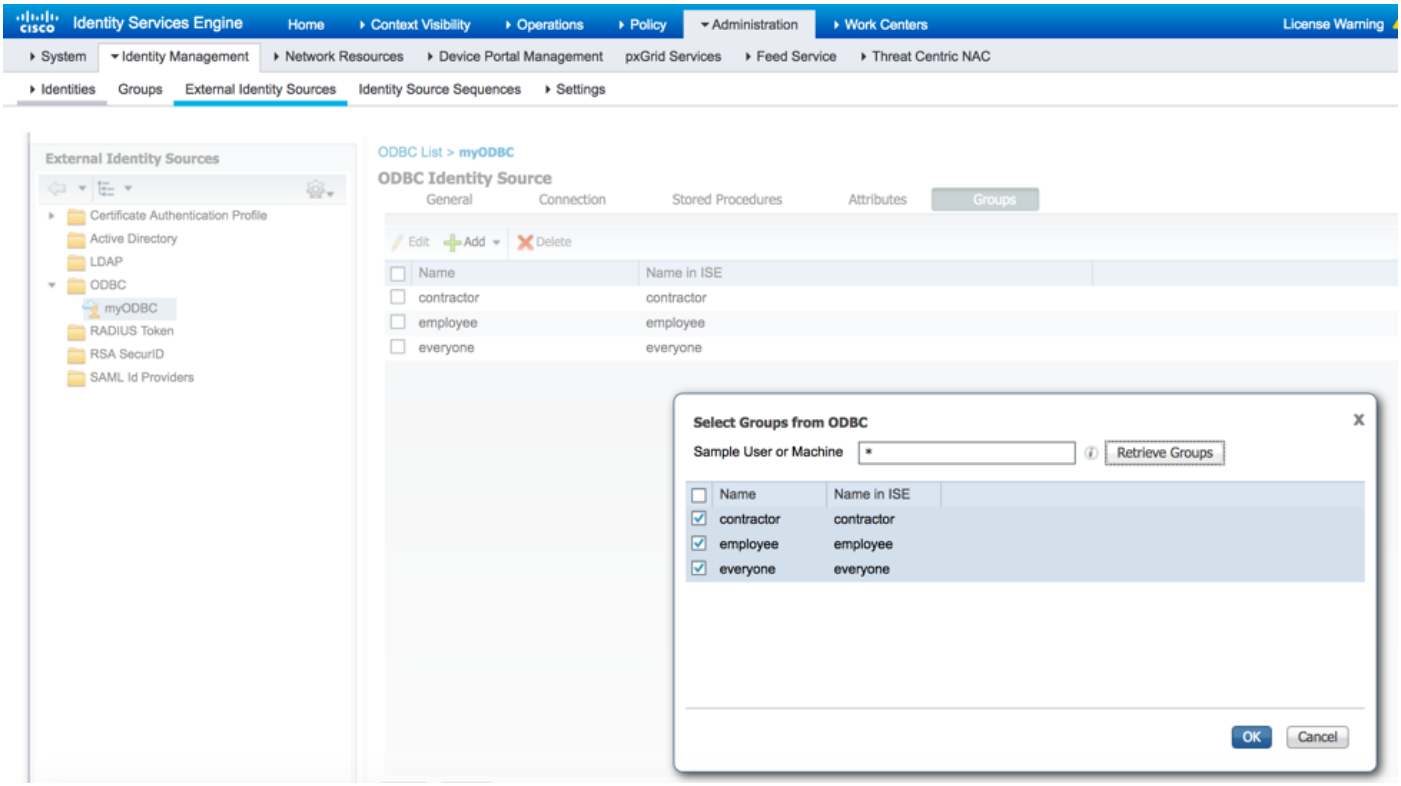
Once done, go back to **Connection** tab and test connection:



Fetch attributes from MySQL, click on **Attributes** tab:

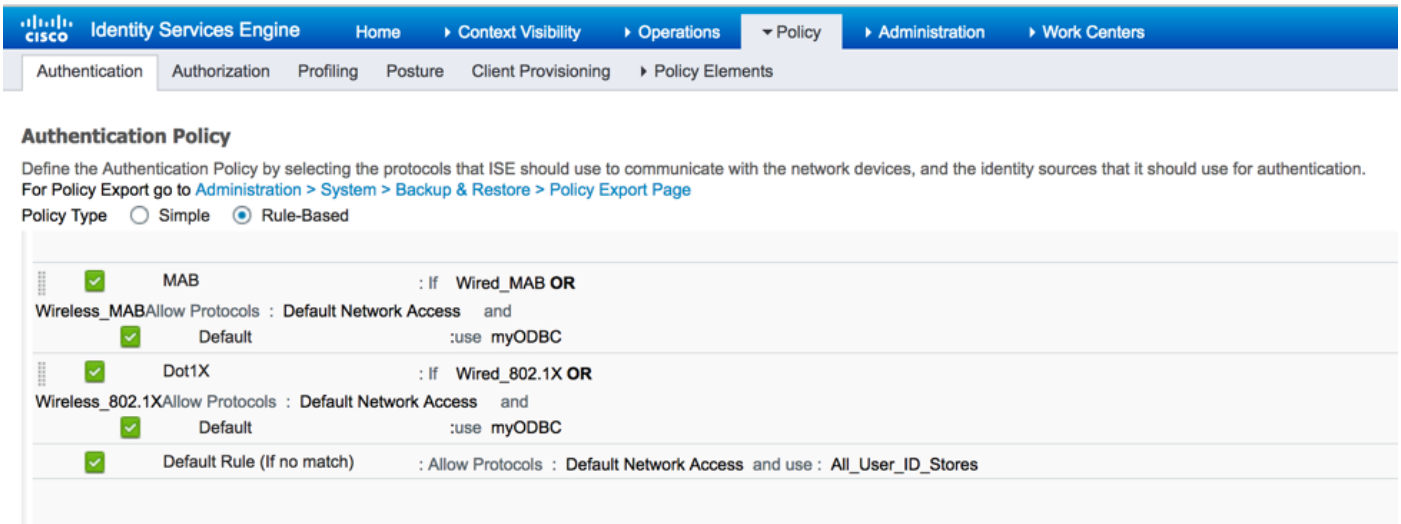


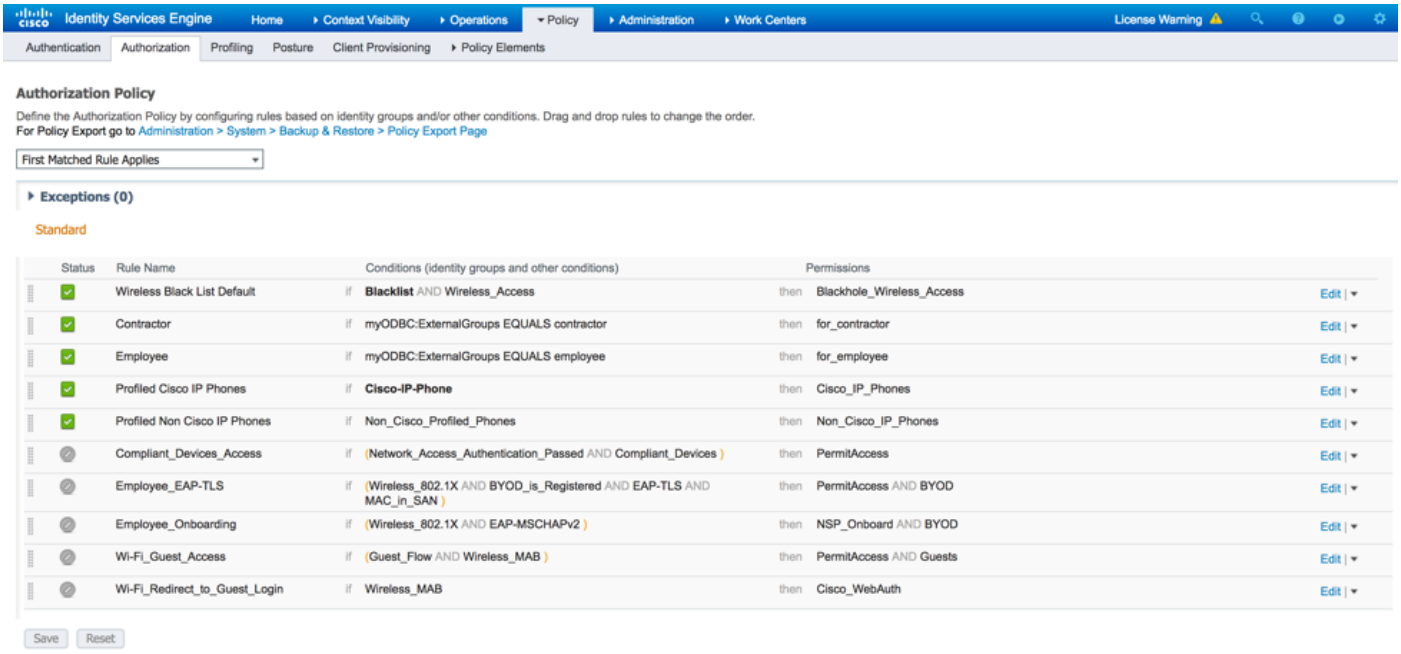
Fetch groups the same way:



5. Configure Authentication and Authorization policies:

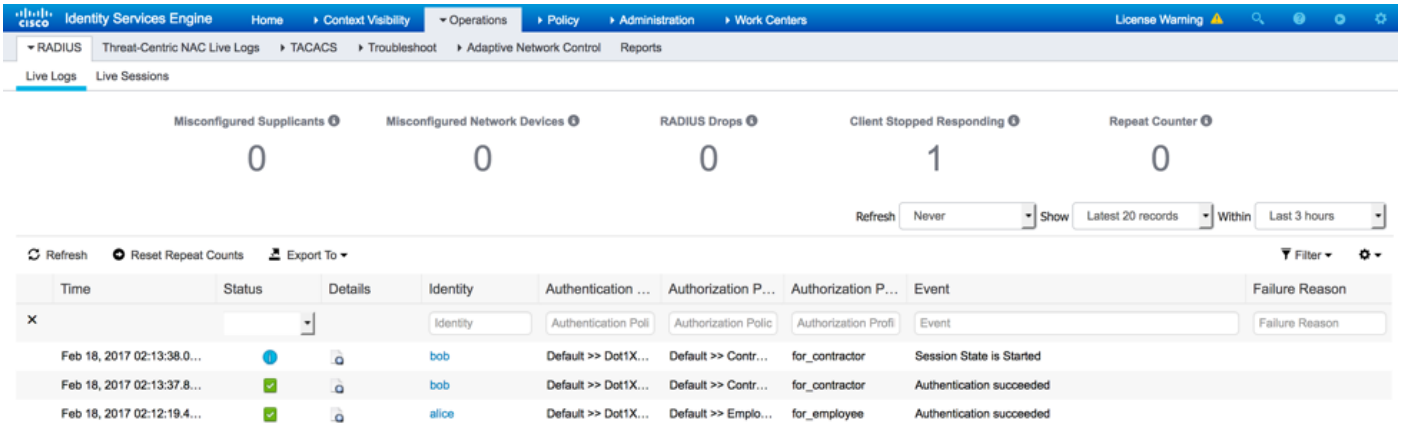
Configure ISE to authenticate and authorize users from MySQL database. Navigate to **Policy > Authentication** and **Policy > Authorization**:





Verify

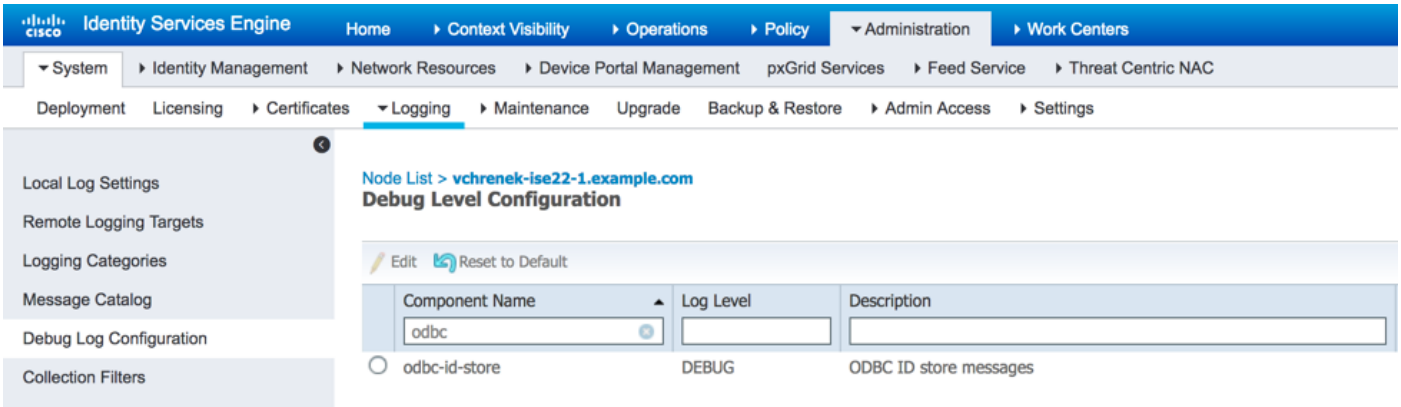
Two authentication flows were tested: PEAP-MSCHAPv2 and MAB. Alice is part of employee group on MySQL, Bob is part of contractor group:



Troubleshoot

Debugs on ISE

In order to enable debugs on ISE, navigate to **Administration > System > Logging > Debug Log Configuration**, select PSN node and change the log level of **odbc-id-store** component to **DEBUG:**



Logs to be checked - prrt-server.log and prrt-management.log. You can tail them directly from CLI of ISE:

```
vchrenek-ise22-1/admin# show logging application prrt-management.log tail
```

During authentication of user bob, ISE has to fetch plain text password and following stored procedure is used **`ISEFetchPasswordReturnsRecordset`**:

```
2017-02-18 14:13:37,565 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcIdStore -::::- ODBC
ID Store Operation: Fetch Plain Text Password. Username=bob, SessionID=0a3e94660000090658a8487f
2017-02-18 14:13:37,566 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.CustomerLog -::::- Write
customer log message: 24861
2017-02-18 14:13:37,567 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
::::- OdbcConnectionPool - get connection
2017-02-18 14:13:37,567 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
::::- OdbcConnectionPool - use existing connection
2017-02-18 14:13:37,568 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
::::- OdbcConnectionPool - connections in use: 1
2017-02-18 14:13:37,568 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnection -::::-
Fetch plain text password
2017-02-18 14:13:37,568 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnection -::::-
Prepare stored procedure call, procname=ISEFetchPasswordReturnsRecordset
2017-02-18 14:13:37,568 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnection -::::-
Using recordset to obtain stored procedure result values
2017-02-18 14:13:37,568 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.CustomerLog -::::- Write
customer log message: 24855
2017-02-18 14:13:37,568 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnection -::::-
Text: {call ISEFetchPasswordReturnsRecordset(?)}
2017-02-18 14:13:37,568 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnection -::::-
Setup stored procedure input parameters, username=bob
2017-02-18 14:13:37,568 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnection -::::-
Execute stored procedure call
2017-02-18 14:13:37,571 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnection -::::-
Process stored procedure results
2017-02-18 14:13:37,571 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnection -::::-
Obtain stored procedure results from recordset
2017-02-18 14:13:37,571 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnection -::::-
Received result recordset, number of columns=5
2017-02-18 14:13:37,571 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnection -::::-
Results successfully parsed from recordset
2017-02-18 14:13:37,572 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
::::- OdbcConnectionPool - release connection
2017-02-18 14:13:37,572 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
::::- OdbcConnectionPool - connections in use: 0
2017-02-18 14:13:37,572 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcIdStore -::::- Call
to ODBC DB succeeded
2017-02-18 14:13:37,572 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.OdbcAuthResult -::::-
Authentication result: code=0, Conection succeeded=false, odbcDbErrorString=no error,
odbcStoredProcedureCustomerErrorString=null, accountInfo=This is a very good user, give him all
```

access, group=11

Since ISE has to check ODBC group assignment, it has to retrieve the groups:

```
2017-02-18 14:13:37,572 DEBUG  [Thread-493][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write
customer log message: 24862
2017-02-18 14:13:37,728 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Username=bob, SessionID=0a3e94660000090658a8487f
2017-02-18 14:13:37,728 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Fetch user groups. Username=bob, SessionID=0a3e94660000090658a8487f
2017-02-18 14:13:37,728 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write
customer log message: 24869
2017-02-18 14:13:37,729 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - get connection
2017-02-18 14:13:37,729 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - use existing connection
2017-02-18 14:13:37,729 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - connections in use: 1
2017-02-18 14:13:37,729 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Fetch user groups
2017-02-18 14:13:37,729 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Prepare stored procedure call, procname=ISEGroups
2017-02-18 14:13:37,729 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Text: {call ISEGroups(?,?)}
2017-02-18 14:13:37,733 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Setup stored procedure input parameters, username=bob
2017-02-18 14:13:37,733 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Execute stored procedure call
2017-02-18 14:13:37,740 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Process stored procedure results
2017-02-18 14:13:37,740 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Received result recordset, total number of columns=1
2017-02-18 14:13:37,740 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
According to column number expect multiple rows (vertical attributes/groups retured result)
2017-02-18 14:13:37,740 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Fetched data: ExternalGroup=everyone
2017-02-18 14:13:37,740 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Fetched data: ExternalGroup=contractor
2017-02-18 14:13:37,740 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Results successfully parsed from recordset
2017-02-18 14:13:37,740 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Result code indicates success
2017-02-18 14:13:37,740 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - release connection
2017-02-18 14:13:37,740 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - connections in use: 0
2017-02-18 14:13:37,740 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- Call
to ODBC DB succeeded
2017-02-18 14:13:37,740 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write
customer log message: 24870
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Got groups...
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Got groups(0) = everyone
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Setting Internal groups(0) = everyone
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Got groups(1) = contractor
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Setting Internal groups(1) = contractor
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Username=bob, ExternalGroups=[everyone, contractor]
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Fetch user attributes. Username=bob, SessionID=0a3e94660000090658a8487f
```

```
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write
customer log message: 24872
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - get connection
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - use existing connection
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - connections in use: 1
```

The same applies for attributes:

```
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Fetch user attributes
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Prepare stored procedure call, procname=ISEAttrsH
2017-02-18 14:13:37,741 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Text: {call ISEAttrsH(?,?)}
2017-02-18 14:13:37,745 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Setup stored procedure input parameters, username=bob
2017-02-18 14:13:37,746 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Execute stored procedure call
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Process stored procedure results
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Received result recordset, total number of columns=3
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
According to column number expect multiple columns (hotizontal attributes/groups retured result)
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Fetched data: eye_color=green
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Fetched data: floor=1
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Fetched data: is_certified=true
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Results successfully parsed from recordset
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Result code indicates success
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - release connection
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - connections in use: 0
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- Call
to ODBC DB succeeded
2017-02-18 14:13:37,749 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write
customer log message: 24873
2017-02-18 14:13:37,750 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user attrs. Username=bob, Setting myODBC.eye_color to green
2017-02-18 14:13:37,750 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user attrs. Username=bob, Setting myODBC.floor to 1
2017-02-18 14:13:37,750 DEBUG  [Thread-259][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user attrs. Username=bob, Setting myODBC.is_certified to true
```

Related information

- **Technical Support & Documentation - Cisco Systems**
- **ISE 2.2 Release Notes**
- **ISE 2.2 Hardware Installation Guide**
- **ISE 2.2 Upgrade Guide**
- **ISE 2.2 Engine Administrator Guide**