Configuración ODBC en el 2.1 ISE con PostgreSQL

Contenido

[Introducción](#)
[prerrequisitos](#)
[Requisitos](#)
[Componentes Utilizados](#)
[Configurar](#)
[Paso 1. Configuración básica de PostgreSQL](#)
[Paso 2. Configuración ISE](#)
[Paso 3. Autenticación de usuario de la configuración](#)
[Paso 4. Extracción del grupo de la configuración](#)
[Paso 5. La configuración atribuye la extracción](#)
[Verificación](#)
[Troubleshooting](#)
[Referencias](#)

Introducción

Este documento describe cómo configurar el Identity Services Engine (ISE) con el servidor de PostgreSQL para la autenticación ISE usando la Conectividad abierta de base de datos (ODBC).

Nota: La autenticación de la Conectividad abierta de base de datos (ODBC) requiere el ISE poder traer una contraseña del usuario del sólo texto. La contraseña se puede cifrar en la base de datos, pero tiene que ser desencriptada por el procedimiento almacenado.

Prerrequisitos

Requisitos

Cisco recomienda que tenga conocimiento sobre estos temas:

- 2.1 del Cisco Identity Services Engine
- Base de datos y conceptos ODBC
- PostgreSQL

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- 2.1 del Identity Services Engine
- Centos 7
- PostgreSQL 9.2

Configurar

> Nota: Código de la invitación SQL en este documento como un ejemplo. Generalmente hay más de una manera de cifrar deseó las funciones y todas tienen sus ventajas y desventajas.

Paso 1. Configuración básica de PostgreSQL

Los pasos para la configuración incluyen la creación de la base de datos y a un usuario para el ISE con los permisos para acceder esa base de datos.

1. Del usuario del postgres cree al usuario del isedb:

```
$ createuser --interactive
Enter name of role to add: isedb
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) y
Shall the new role be allowed to create more new roles? (y/n) n
Password:
```

2. Cree una base de datos

```
$ createdb isedb
```

o con el SQL:

```
CREATE DATABASE isedb WITH TEMPLATE = template0 OWNER = isedb;
REVOKE ALL ON DATABASE isedb FROM PUBLIC;
REVOKE ALL ON DATABASE isedb FROM postgres;
GRANT CONNECT,TEMPORARY ON DATABASE isedb TO PUBLIC;
GRANT ALL ON DATABASE isedb TO isedb;
```

3. Permita el acceso a PostgreSQL

```
sudo vi /var/lib/pgsql/data/pg_hba.conf
```

Encuentre las líneas que parece el, cerca de la parte inferior del archivo:

```
host all all 127.0.0.1/32 ident
host all all ::1/128 ident
```

Entonces substituya la **identificación** por el **md5**, así que parecen esto:

```
host all all 127.0.0.1/32 md5
host all all 10.0.0.0/8 md5
```

4. Permita las conexiones remotas a PgSQL

Usted necesita abrir el archivo de configuración **/var/lib/pgsql/data/postgresql.conf de** PostgreSQL. Línea de configuración del hallazgo que lee:

```
listen_addresses='localhost'
```

y cambio a

```
listen_addresses='*'
```

Permita las conexiones de todos los direccionamientos. Línea de la configuración del puerto de Uncomment (si está comentado):

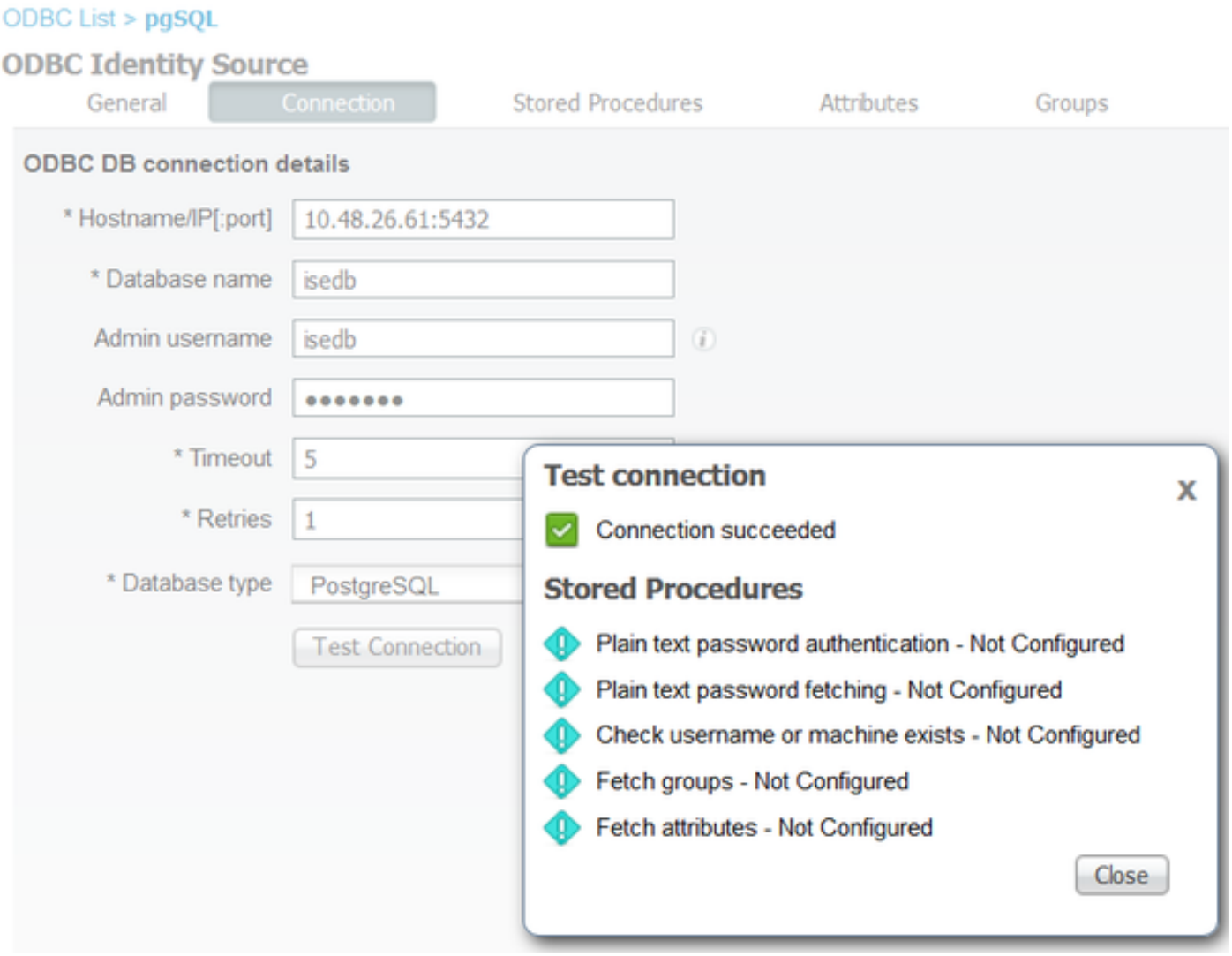
```
port = 5432
```

5. Reinicio PgSQL:

```
$ sudo systemctl start postgresql
$ sudo systemctl enable postgresql
```

Paso 2. Configuración ISE

Cree una fuente de la identidad ODBC en la **administración > fuente externa de la identidad > ODBC** y conexión de prueba:



Paso 3. Autenticación de usuario de la configuración

La autenticación ISE al ODBC utiliza los procedimientos almacenados. Es posible seleccionar el tipo de procedimientos. En este ejemplo utilizamos los parámetros como vuelta. Para otros procedimientos, refiera a la [guía de administración del 2.1 del Cisco Identity Services Engine]

Consejo: Es posible volver los parámetros Nombrados en vez del resultset. Es apenas un tipo diferente de salida, las funciones es lo mismo.

1. Cree la tabla. Aseegurese le fijan las configuraciones de la identidad en la **Clave primaria**

```
CREATE TABLE "ISE_Users" (
user_id uuid NOT NULL,
username character varying NOT NULL,
password character varying NOT NULL
);


ALTER TABLE public."ISE_Users" OWNER TO isedb;
ALTER TABLE ONLY "ISE_Users"
ADD CONSTRAINT "ISE_Users_pkey" PRIMARY KEY (user_id);
```

2. Funcione con esta interrogación para insertar a un usuario

```
INSERT INTO "ISE_Users" VALUES ('8cc4b9b9-117a-46c4-879e-d764c9685e80', 'user1', 'password1');
```

O

```
INSERT INTO "ISE_Users" VALUES (uuid_generate_v1(), 'user1', 'password1');
```

Y aprenda y salve UUID generado de un usuario nuevo con esta interrogación

```
SELECT user_id FROM "ISE_Users" WHERE username = 'user1';
```

3. Cree un procedimiento para la autenticación de contraseña del sólo texto (usada para el método interno PAP, EAP-GTC, el TACACS)

```
CREATE FUNCTION iseauthuserplainreturnsparameters(ise_username text, ise_password text, OUT
result integer, OUT ise_group text, OUT acctinfo text, OUT errorstring text) RETURNS record
LANGUAGE plpgsql IMMUTABLE SECURITY DEFINER
AS $$
DECLARE
c int;
BEGIN
select count(*) into c from "ISE_Users" where username = ise_username and password =
ise_password;
IF c > 0 THEN
result := 0;
ise_group := cast ('11' as text);
acctinfo := cast ('This is a very good user, give him all access' as text);
errorstring := cast ('No error' as text);
else
result := 3;
ise_group := cast ('11' as text);
acctinfo := cast ('User is unknown or invalid password' as text);
errorstring := cast ('User is unknown or invalid password' as text);
END IF;
END;
$$;

ALTER FUNCTION public.iseauthuserplainreturnsparameters(ise_username text, ise_password text,
OUT result integer, OUT ise_group text, OUT acctinfo text, OUT errorstring text) OWNER TO isedb;
```

4. Cree un procedimiento para la recogida de la contraseña del sólo texto (usada para la GRIETA, MSCHAPv1/v2, EAP-MD5, SALTO, método interno del EAP MSCHAPv2, el TACACS)

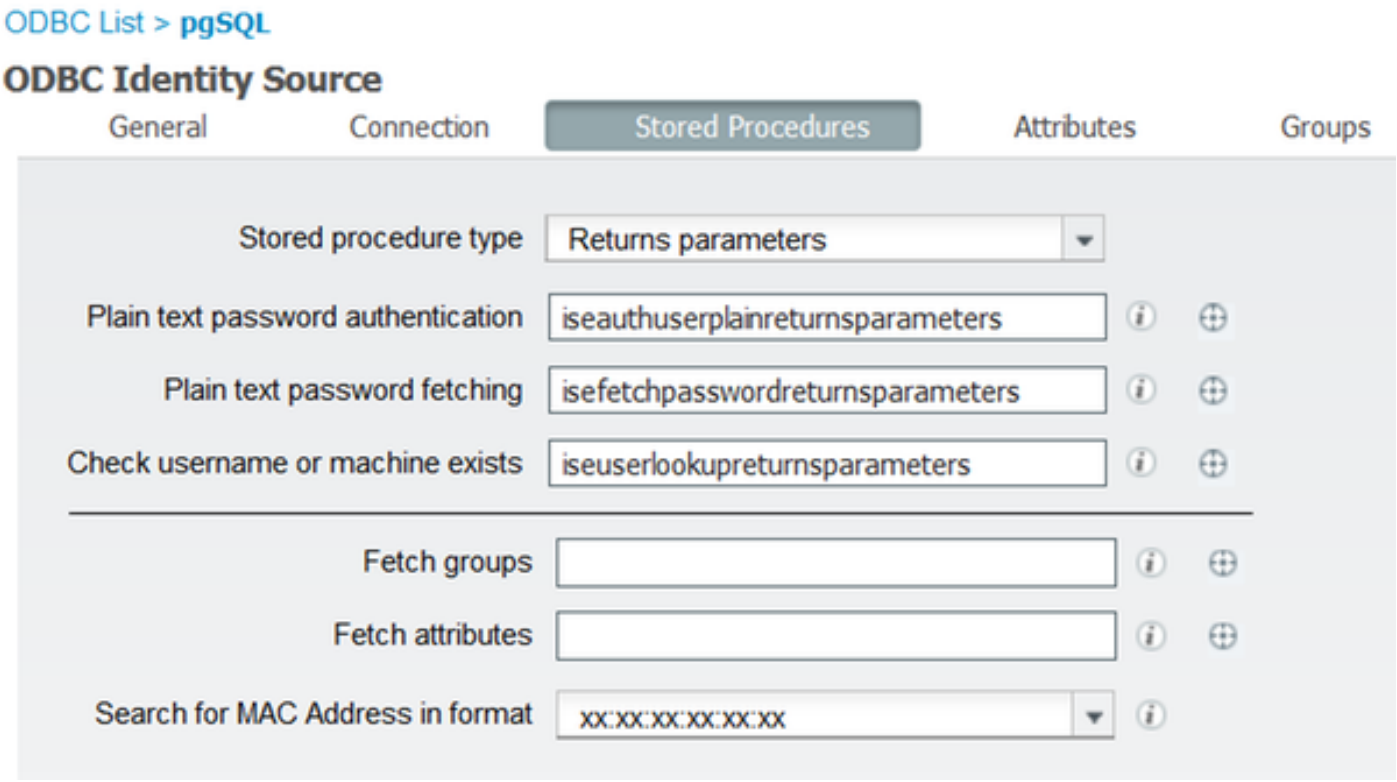
```
CREATE FUNCTION isefetchpasswordreturnsparameters(ise_username text, OUT result integer, OUT
ise_group text, OUT acctinfo text, OUT errorstring text, OUT ise_password text) RETURNS record
LANGUAGE plpgsql IMMUTABLE SECURITY DEFINER
AS $$
DECLARE
c int;
BEGIN
```

```
select count(*) into c from "ISE_Users" where username = ise_username;
IF c > 0 THEN
result := 0;
ise_group := cast ('11' as text);
acctinfo := cast ('This is a very good user, give him all access' as text);
errorstring := cast ('no error' as text);
select password into ise_password from "ISE_Users" where username = ise_username;
else
result := 3;
ise_group := cast ('11' as text);
acctinfo := cast ('User is unknown' as text);
errorstring := cast ('User is unknown' as text);
END IF;
END;
$$;


ALTER FUNCTION public.isefetchpasswordreturnsparameters(ise_username text, OUT result integer,
OUT ise_group text, OUT acctinfo text, OUT errorstring text, OUT ise_password text) OWNER TO
isedb;
```

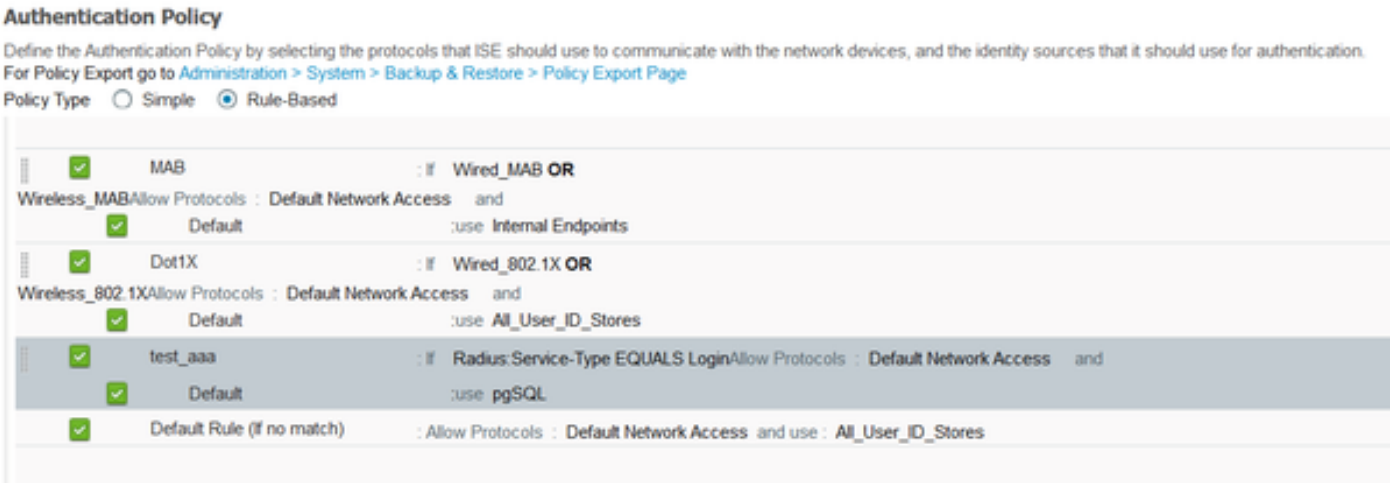
5. Cree un procedimiento para el nombre de usuario del control o la máquina existe (utilizado para el MAB, rápido vuelva a conectar del PEAP, del EAP-FAST y del EAP-TTLS)

```
CREATE FUNCTION iseuserlookupreturnsparameters(ise_username text, OUT result integer, OUT
ise_group text, OUT acctinfo text, OUT errorstring text) RETURNS record
LANGUAGE plpgsql IMMUTABLE SECURITY DEFINER
AS $$
DECLARE
c int;
BEGIN
select count(*) into c from "ISE_Users" where username = ise_username;
IF c > 0 THEN
result := 0;
ise_group := cast ('11' as text);
acctinfo := cast ('good user' as text);
errorstring := cast ('no error' as text);
else
result := 3;
ise_group := cast ('11' as text);
acctinfo := cast ('bad user' as text);
errorstring := cast ('bad password' as text);
END IF;
END;
$$;


ALTER FUNCTION public.iseuserlookupreturnsparameters(ise_username text, OUT result integer, OUT
ise_group text, OUT acctinfo text, OUT errorstring text) OWNER TO isedb;
```

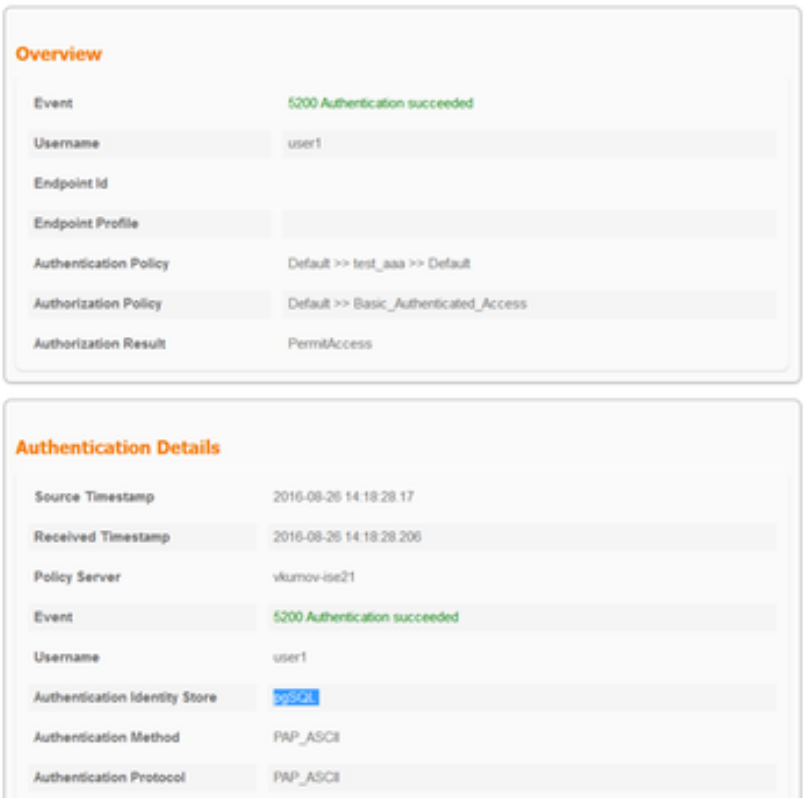
6. Configure los procedimientos en el ISE y sálvelos



7. Cree una regla de la autenticación simple usando el ODBC y pruébela




```
BAHAMUT#test aaa group ISE user1 password1 legacy
Attempting authentication test to server-group ISE using radius
User was successfully authenticated.
```



Paso 4. Extracción del grupo de la configuración

1. Cree las tablas que contienen a los grupos de usuarios y otras usadas para la asignación múltiple

```
CREATE TABLE "Groups" (
group_id uuid NOT NULL,
group_name character varying(255) NOT NULL,
group_description text
);

ALTER TABLE public."Groups" OWNER TO isedb;

ALTER TABLE ONLY "Groups"
ADD CONSTRAINT "Groups_pkey" PRIMARY KEY (group_id);

CREATE TABLE "User_Groups_Mapping" (
user_id uuid,
group_id uuid
);

ALTER TABLE public."User_Groups_Mapping" OWNER TO isedb;

ALTER TABLE ONLY "User_Groups_Mapping"
ADD CONSTRAINT "User_Groups_Mapping_group_id_fkey" FOREIGN KEY (group_id) REFERENCES
"Groups"(group_id) ON UPDATE CASCADE ON DELETE CASCADE;

ALTER TABLE ONLY "User_Groups_Mapping"
ADD CONSTRAINT "User_Groups_Mapping_user_id_fkey" FOREIGN KEY (user_id) REFERENCES
"ISE_Users"(user_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

2. Agregue los grupos y las asignaciones, de modo que el user1 pertenezca a dos grupos

```
INSERT INTO "Groups" VALUES ('f7dfee5c-bd06-4703-9de0-4d334ea5ec02', 'Admins', 'Group for
administrators');
INSERT INTO "Groups" VALUES ('51fc0ccd-caf8-4585-ba20-6596948c879d', 'Users', 'Group for
users');
```

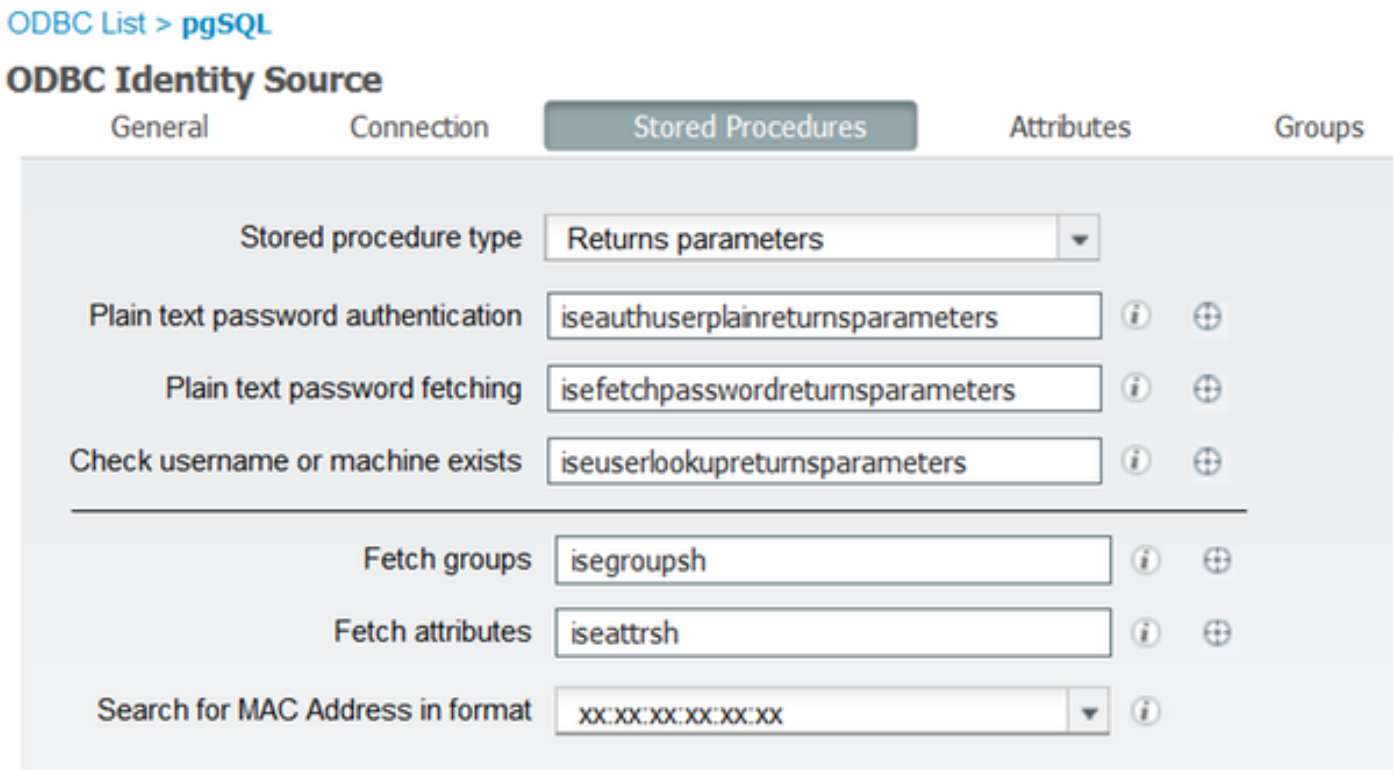
```
INSERT INTO "Groups" VALUES ('7b7e72bc-ea22-470c-8578-1dd86b1a1843', 'Laptops', 'Group for users
with laptops');

INSERT INTO "User_Groups_Mapping" VALUES ('8cc4b9b9-117a-46c4-879e-d764c9685e80', 'f7dfee5c-
bd06-4703-9de0-4d334ea5ec02');
INSERT INTO "User_Groups_Mapping" VALUES ('8cc4b9b9-117a-46c4-879e-d764c9685e80', '7b7e72bc-
ea22-470c-8578-1dd86b1a1843');
```

O genere nuevo UUIDs, no obstante usted necesitará aprenderlos con las interrogaciones SELECTAS.

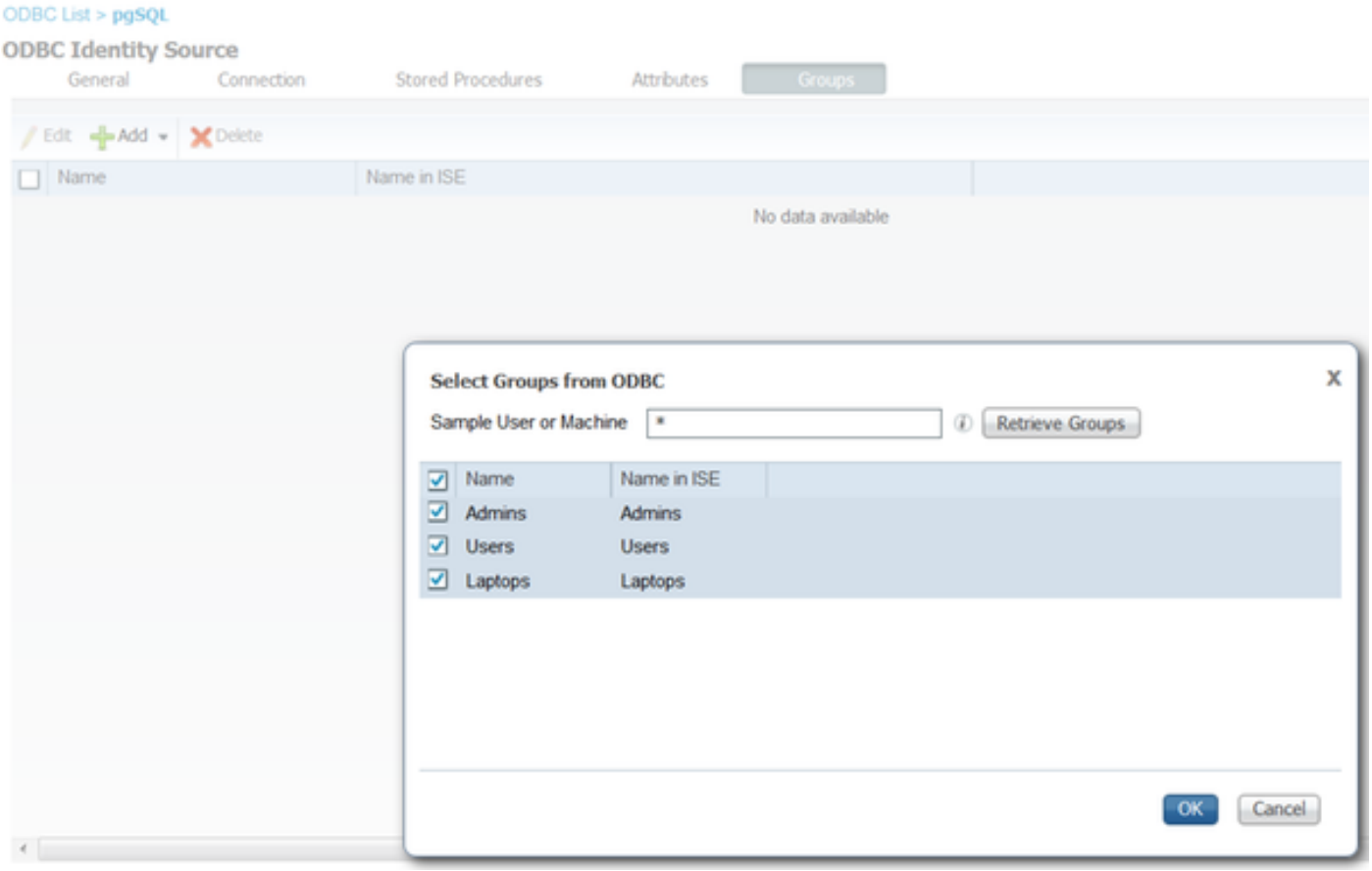
3. Cree el tipo de valor devuelto y un procedimiento de la extracción del grupo

```
CREATE TYPE g4type AS (
result integer,
group_n text
);

ALTER TYPE public.g4type OWNER TO isedb;

CREATE FUNCTION isegroupsh(ise_username text) RETURNS SETOF g4type
LANGUAGE plpgsql IMMUTABLE SECURITY DEFINER
AS $$
DECLARE
c int;
i int;
r g4type%rowtype;
BEGIN
if ise_username = '*' then
for r in select 0, cast(group_name as text) from "Groups"
loop
return next r;
end loop;
else
select count(*) into c from "ISE_Users" where username = ise_username;
IF c > 0 THEN
for r in select 0, cast(group_name as text) from "Groups" where group_id in (
select group_ID from "User_Groups_Mapping" where "User_Groups_Mapping".user_id IN (
select user_id from "ISE_Users" where username = ise_username
) )
loop
return next r;
end loop;
else
return query select 1,cast ('' as text);
END IF;
end if;
END;
$$;

ALTER FUNCTION public.isegroupsh(ise_username text) OWNER TO isedb;
```

4. Asocíela para traer a los grupos



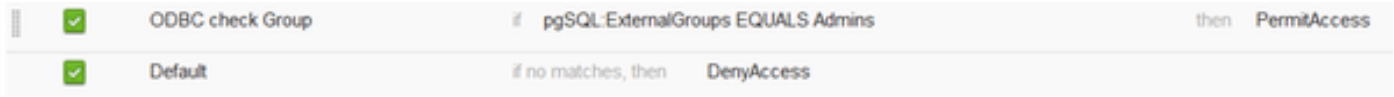
5. Traiga a los grupos y agreguelos en la **fuente de la identidad ODBC**



6. Agregue a otro usuario que no pertenezca a cualquier grupo

```
INSERT INTO "ISE_Users" VALUES ('592136bb-9c47-49ff-8eca-9adfb2016b1c', 'user2', 'password2');
```

7. Cree una **directiva de la autorización de la** prueba y pruébela




```
BAHAMUT#test aaa group ISE user1 password1 legacy
Attempting authentication test to server-group ISE using radius
User was successfully authenticated.

BAHAMUT#test aaa group ISE user2 password2 legacy
Attempting authentication test to server-group ISE using radius
User authentication request was rejected by server.
```

SelectedAuthenticationIdentityStores	pgSQL
AuthorizationPolicyMatchedRule	ODBC check Group
CPMSessionID	0a301a321uM9iabemtwC3JmOxM0PEPNRCy44aEudtrNg2ajmJGg
ISEPolicySetName	Default
AllowedProtocolMatchedRule	test_aaa
IdentitySelectionMatchedRule	Default
Network Device Profile	Cisco
Location	Location#All Locations
Device Type	Device Type#All Device Types
ExternalGroups	Admins
ExternalGroups	Laptops
RADIUS Username	user1

Paso 5. La configuración atribuye la extracción

1. Para simplificar este ejemplo, una tabla plana se utiliza para los atributos

```
CREATE TABLE "User_Attributes" (
user_id uuid,
attribute_name character varying(255),
attribute_value character varying(255)
);

ALTER TABLE public."User_Attributes" OWNER TO isedb;

ALTER TABLE ONLY "User_Attributes"
ADD CONSTRAINT "User_Attributes_user_id_fkey" FOREIGN KEY (user_id) REFERENCES
"ISE_Users"(user_id) ON UPDATE CASCADE ON DELETE CASCADE;
```

2. Cree un atributo para ambos usuarios

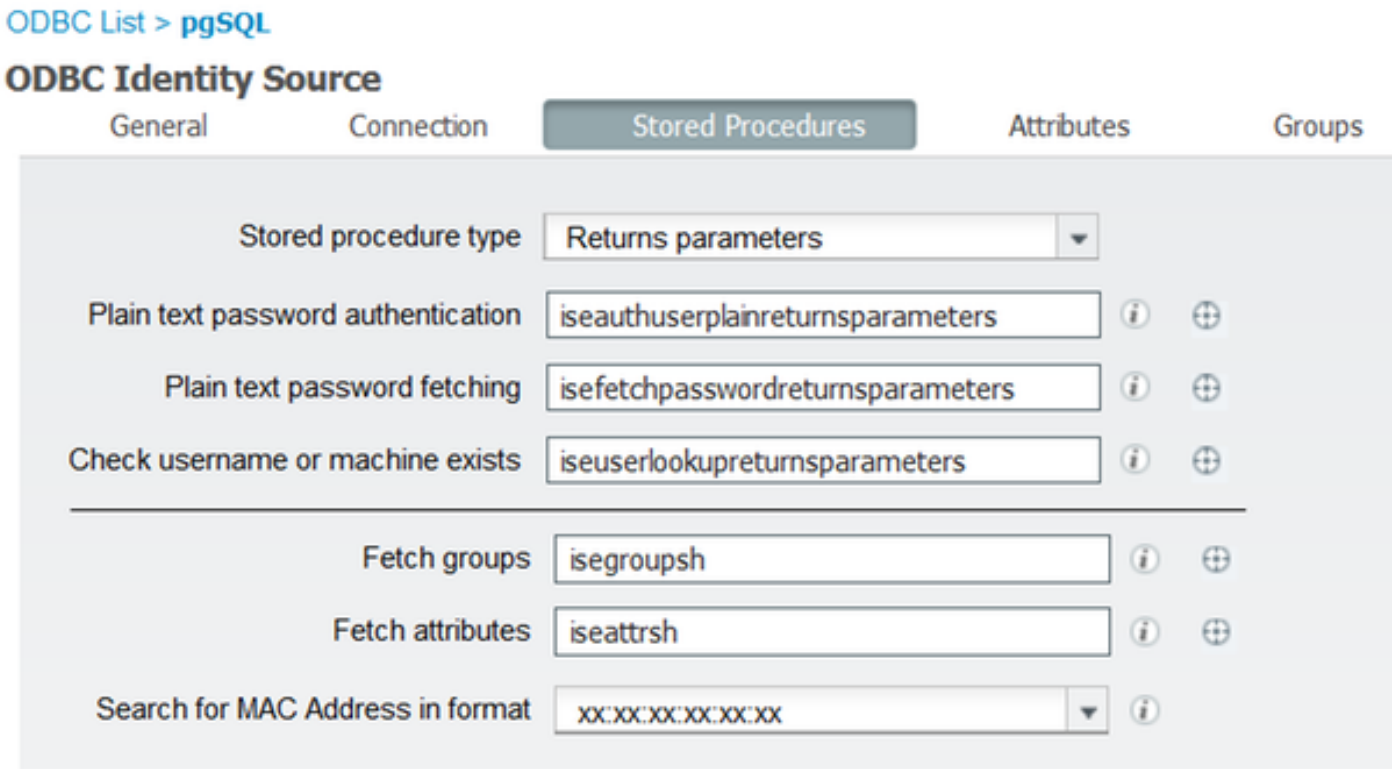
```
INSERT INTO "User_Attributes" VALUES ('8cc4b9b9-117a-46c4-879e-d764c9685e80', 'SecurityLevel',
'10');
```

```
INSERT INTO "User_Attributes" VALUES ('592136bb-9c47-49ff-8eca-9adfb2016b1c', 'SecurityLevel',
'5');
INSERT INTO "User_Attributes" VALUES ('592136bb-9c47-49ff-8eca-9adfb2016b1c', 'IdleTimeout',
'5');
```

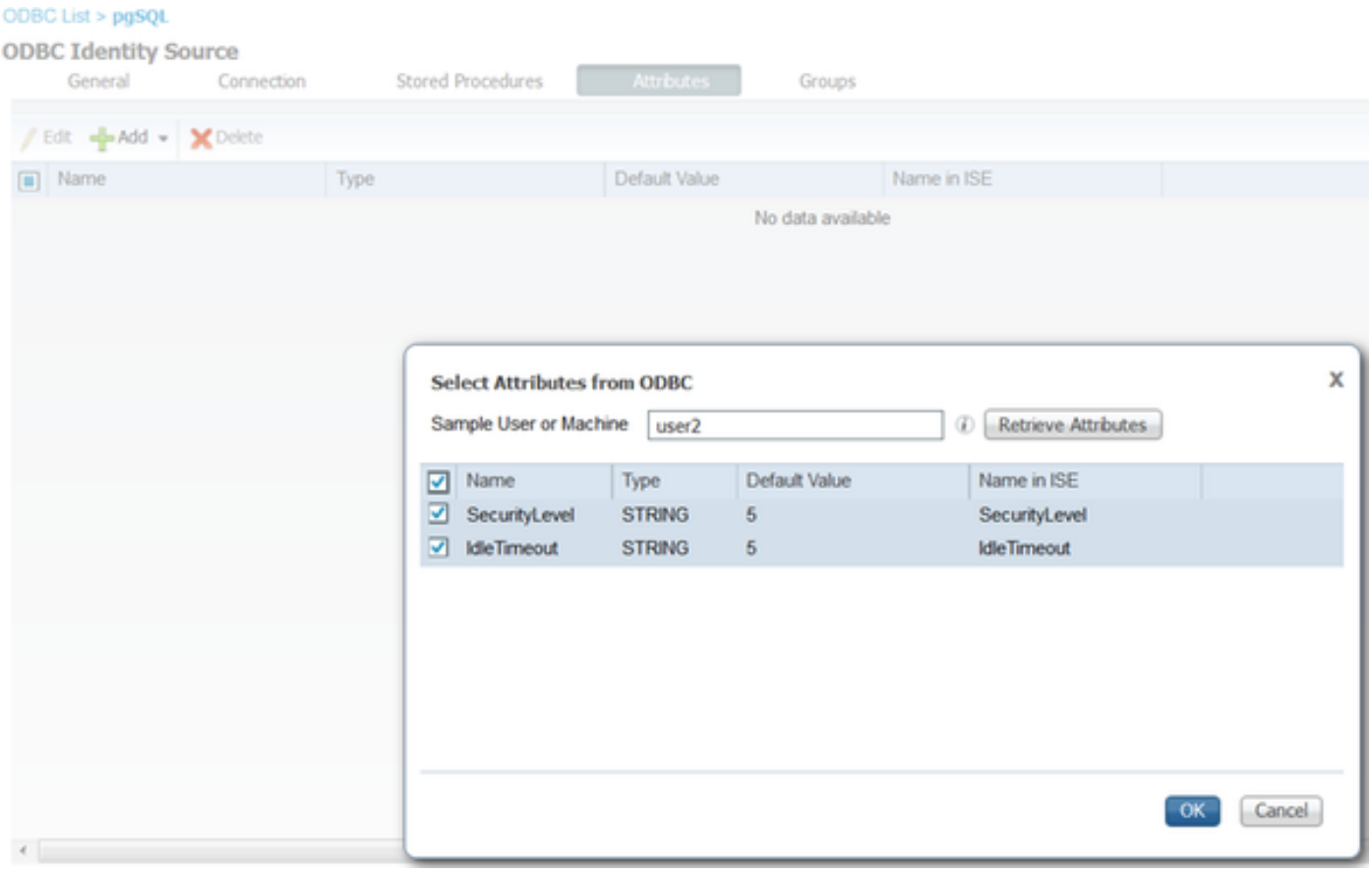
3. Cree un **tipo de valor devuelto** y un **procedimiento almacenado**

```
CREATE TYPE a4type AS (
result integer,
attr_name text,
attr_value text
);

ALTER TYPE public.a4type OWNER TO isedb;

CREATE FUNCTION iseattrsh(ise_username text) RETURNS SETOF a4type
LANGUAGE plpgsql IMMUTABLE SECURITY DEFINER
AS $$
DECLARE
c int;
r a4type%rowtype;
BEGIN
select count(*) into c from "ISE_Users" where username = ise_username;
IF c > 0 THEN
for r in select 0, cast(s.attribute_name as text), cast(s.attribute_value as text) from
"User_Attributes" as s where user_id in(SELECT user_id from "ISE_Users" where username =
ise_username)
loop
return next r;
end loop;
else
return query select 1, cast ('' as text);
END IF;
END;
$$;

ALTER FUNCTION public.iseattrsh(ise_username text) OWNER TO isedb;
```

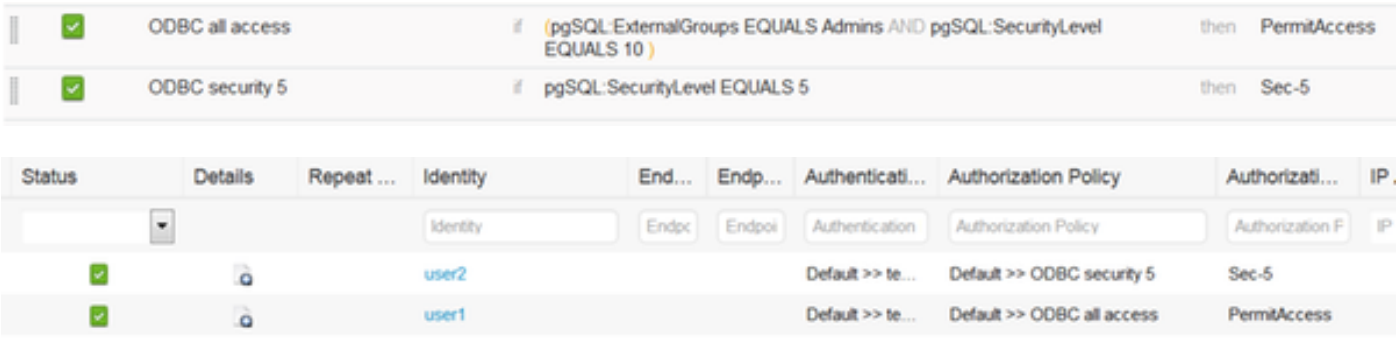
4. Asocíelo **para traer los atributos**



5. Traiga los atributos



6. Ajuste las directivas ISE y pruébelas



Verificación

Usted debe ahora poder autenticar a los usuarios contra el ODBC y extraer sus grupos y atributos.

Ejemplo:

Troubleshooting

Si la conexión no es acertada en la **cola de prrt-management.log de la aplicación del** comando show logging del uso ISE mientras que intenta conectar.

Ejemplo de las credenciales incorrectas:

```
2016-08-28 13:55:47,017 WARN [admin-http-pool1372][] cisco.cpm.odbcidstore.impl.PostgresDbAccess
-:admin::- Connection to ODBC DB failed. Exception: org.postgresql.util.PSQLException: FATAL:
password authentication failed for u
```

```
ser "isedb_wrong"
org.postgresql.util.PSQLException: FATAL: password authentication failed for user "isedb_wrong"
at org.postgresql.Driver$ConnectThread.getResult(Driver.java:365)
at org.postgresql.Driver.connect(Driver.java:288)
at java.sql.DriverManager.getConnection(DriverManager.java:664)
at java.sql.DriverManager.getConnection(DriverManager.java:208)
at com.cisco.cpm.odbcidstore.impl.PostgresDbAccess.connect(PostgresDbAccess.java:46)
at com.cisco.cpm.odbcidstore.impl.OdbcConnection.connect(OdbcConnection.java:72)
at com.cisco.cpm.odbcidstore.impl.OdbcIdStore.performTest(OdbcIdStore.java:377)
at
com.cisco.cpm.odbcidstore.impl.OdbcIdStore.testConnectionAndConfiguration(OdbcIdStore.java:469)
at
com.cisco.cpm.odbcidstore.impl.OdbcIdStoreManager.testConnectionAndConfiguration(OdbcIdStoreMana
ger.java:84)
at com.cisco.cpm.admin.ac.actions.ODBCLPInputAction.testConnection(ODBCLPInputAction.java:749)
```

Ejemplo del nombre incorrecto DB:

```
2016-08-28 13:53:43,174 WARN [admin-http-pool1372][] cisco.cpm.odbcidstore.impl.PostgresDbAccess
-:admin::- Connection to ODBC DB failed. Exception: org.postgresql.util.PSQLException: FATAL:
database "isedb_wrong" does not exis
t
org.postgresql.util.PSQLException: FATAL: database "isedb_wrong" does not exist
at org.postgresql.Driver$ConnectThread.getResult(Driver.java:365)
at org.postgresql.Driver.connect(Driver.java:288)
at java.sql.DriverManager.getConnection(DriverManager.java:664)
at java.sql.DriverManager.getConnection(DriverManager.java:208)
at com.cisco.cpm.odbcidstore.impl.PostgresDbAccess.connect(PostgresDbAccess.java:46)
at com.cisco.cpm.odbcidstore.impl.OdbcConnection.connect(OdbcConnection.java:72)
at com.cisco.cpm.odbcidstore.impl.OdbcIdStore.performTest(OdbcIdStore.java:377)
at
com.cisco.cpm.odbcidstore.impl.OdbcIdStore.testConnectionAndConfiguration(OdbcIdStore.java:469)
at
com.cisco.cpm.odbcidstore.impl.OdbcIdStoreManager.testConnectionAndConfiguration(OdbcIdStoreMana
ger.java:84)
at com.cisco.cpm.admin.ac.actions.ODBCLPInputAction.testConnection(ODBCLPInputAction.java:749)
```

Para resolver problemas las operaciones DB, ODBC-identificación-**almacén de los** componentes del registro del permiso al nivel de debug bajo la **administración > el sistema > configuración del registro del registro > del debug**.

Los registros se colocan en el archivo de **prrt-management.log**.

Ejemplo para el **user1**:

```
2016-08-28 14:01:01,116 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Authenticate Plain Text Password. Username=user1,
SessionID=0a301a32OuqzqoKTrY02KoCjdWN6PlZtBX1/vhDXxN9nQTBFM8g
2016-08-28 14:01:01,118 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.CustomerLog -:::-
Write customer log message: 24852
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - get connection
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - use existing connection
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - connections in use: 1
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Authenticate plain text password
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Prepare stored procedure call, procname=iseauthuserplainreturnsparameters
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
```

Using output parameters to obtain stored procedure result values
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.CustomerLog -:::-
Write customer log message: 24856
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Text: {call iseauthuserplainreturnsparameters(?, ?, ?, ?, ?, ?)}
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Setup stored procedure input parameters, username=user1, password=***
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Setup stored procedure output parameters
2016-08-28 14:01:01,119 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Execute stored procedure call
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Process stored procedure results
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Obtain stored procedure results from output parameters
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Results successfully parsed from output parameters
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - release connection
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - connections in use: 0
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- Call
to ODBC DB succeeded
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.OdbcAuthResult -:::-
Authentication result: code=0, Conection succeeded=false, odbcDbErrorString=No error,
odbcStoredProcedureCustomerErrorString=null, ac
countInfo=This is a very good user, give him all access, group=11
2016-08-28 14:01:01,121 DEBUG [Thread-26349][] cisco.cpm.odbcidstore.impl.CustomerLog -:::-
Write customer log message: 24853
2016-08-28 14:01:01,129 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: **Get all user groups**. Username=**user1**,
SessionID=0a301a32OuqzqoKTrY02KoCjdWN6PlZtBX1/vhDXxN9nQTBFM8g
2016-08-28 14:01:01,131 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: **Fetch user groups**. Username=**user1**,
SessionID=0a301a32OuqzqoKTrY02KoCjdWN6PlZtBX1/vhDXxN9nQTBFM8g
2016-08-28 14:01:01,131 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write
customer log message: 24869
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - get connection
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - use existing connection
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - connections in use: 1
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Fetch user groups
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Prepare stored procedure call, procname=**isegroupsh**
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Text: {call isegroupsh(?)}
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Setup stored procedure input parameters, username=user1
2016-08-28 14:01:01,132 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Execute stored procedure call
2016-08-28 14:01:01,134 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Process stored procedure results
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Received result recordset, total number of columns=2
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
POSTGRES case, first column holds the result param value
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
According to column number expect multiple rows (vertical attributes/groups retured result)
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Fetched data: **ExternalGroup=Admins**
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-

Fetched data: **ExternalGroup=Laptops**
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Results successfully parsed from recordset
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Result code indicates success
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - release connection
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - connections in use: 0
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- Call
to ODBC DB succeeded
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write
customer log message: 24870
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Got groups...
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Got groups(0) = Admins
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Setting Internal groups(0) = Admins
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Got groups(1) = Laptops
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Setting Internal groups(1) = Laptops
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user groups. Username=user1, ExternalGroups=[Admins, Laptops]
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: **Fetch user attributes**. Username=**user1**,
SessionID=0a301a32OuqzqoKTrY02KoCjdWN6PlZtBX1/vhDXxN9nQTBFM8g
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write
customer log message: 24872
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - get connection
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - use existing connection
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - connections in use: 1
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Fetch user attributes
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Prepare stored procedure call, procname=**iseattrsh**
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Text: {call iseattrsh(?)}
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Setup stored procedure input parameters, username=user1
2016-08-28 14:01:01,135 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Execute stored procedure call
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Process stored procedure results
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Received result recordset, total number of columns=3
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
POSTGRES case, first column holds the result param value
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
According to column number expect multiple rows (vertical attributes/groups retured result)
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Fetched data: SecurityLevel=10
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Results successfully parsed from recordset
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnection -:::-
Result code indicates success
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - release connection
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcConnectionPool -
:::- OdbcConnectionPool - connections in use: 0

```
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- Call
to ODBC DB succeeded
2016-08-28 14:01:01,140 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.CustomerLog -:::- Write
customer log message: 24873
2016-08-28 14:01:01,141 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user attrs. Username=user1, Setting pgSQL.SecurityLevel to 10
2016-08-28 14:01:01,141 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user attrs. Username=user1, Setting IdleTimeout to default value : 5
2016-08-28 14:01:01,141 DEBUG [Thread-3076][] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- ODBC
ID Store Operation: Get all user attrs. Username=user1, Setting pgSQL.IdleTimeout to 5
```

Referencias

- Guía de administración del 2.1 del Cisco Identity Services Engine - Configuración ODBC
- 2.1 de la configuración ISE con MS SQL usando el ODBC
- PostgreSQL: Documentación