Configure ISE 2.2 for integration with MySQL server

Contents

Introduction
Prerequisites
Requirements
Components Used
Background information
Configure
Network Diagram
Configurations
1. Configure MySQL on Ubuntu:
2. Configure database and tables:
3. Configure stored procedures
4. Integrate ISE with MySQL:
5. Configure Authentication and Authorization policies:
Verify
Troubleshoot
Debugs on ISE
Related information

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
Background information

ISE 2.2 supports multiple ODBC external sources, one of them is MySQL. You can use ODBC as an 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 of database engines supported in ISE 2.2:

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


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

1. Configure MySQL on Ubuntu:

Update your system:

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

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

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

To access MySQL database:

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

2. Configure database and tables:

Create database:

```bash
mysql> CREATE DATABASE demo_db;
Query OK, 1 row affected (0.00 sec)
```

```bash
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:

```bash
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:

```bash
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:

```bash
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):

```bash
mysql>
mysql> select * from users;
```
Create table of groups:

```
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:

```
INSERT INTO groups  
VALUES  
(1, "everyone");
```

Query OK, 1 row affected (0.00 sec)

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

```
SELECT * FROM groups;
```

Create table for mappings between users and groups

```
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`  
  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

```
INSERT INTO user_group  
VALUES;
```
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 careful 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 14:13:37,568 DEBUG [Thread-493] cisco.cpm.odbcidstore.impl.OdbcConnection - Using recordset to obtain stored procedure result values
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.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, Connection succeeded=false, odbcDbErrorString=no error, odbcStoredProcedureCustomerErrorString=null, accountInfo=This is a very good user, give him all
Since ISE has to check ODBC group assignment, it has to retrieve the groups:

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 14:13:37,740 DEBUG [Thread-259] cisco.cpm.odbcidstore.impl.OdbcIdStore -:::- Call to ODBC DB succeeded
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. Setting Internal groups(1) = contractor

Fetch user groups

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)

Fetched data: ExternalGroup=everyone

Fetched data: ExternalGroup=contractor

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. Setting Internal groups(1) = contractor
The same applies for attributes:

- **Fetch user attributes**
- **Prepare stored procedure call**, procname=ISEAttrsH
- **Text**: \{call ISEAttrsH(?,?)\}
- **Setup stored procedure input parameters**, username=bob
- **Execute stored procedure call**
- **Process stored procedure results**
- **Received result recordset**, total number of columns=3
- According to column number expect multiple columns (horizontal attributes/groups retured result)
- **Fetched data**: eye_color=green
- **Fetched data**: floor=1
- **Fetched data**: is_certified=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**