CA Identity Manager - 14.1

Manage Active Directory Authentication Module

Date: 31-May-2018

Table of Contents

Encrypting ADMINPWD and KEYSTOREPWD 7

Manage Active Directory Authentication Module

By default, CA Identity Manager comes with an out-of-the-box authentication module. This module authenticates the user against the directory that is configured for their environment. The user can also be authenticated to an external Active Directory using the following procedure. You can also encrypt ADMINPWD and KEYSTOREPWD instead of leaving them as clear text.

Note: The Active Directory endpoint must be provisioned by CA Identity Manager so that the Active Directory accounts are synchronized with the CA Identity Manager user store. This procedure also assumes that the administrator is proficient with Active Directory. If you configure the Active Directory authentication model, user password sets from the Forgotten Password or Reset Password tasks automatically propagate to the Active Directory server. This requires an LDAPS connection between the CA Identity Manager Server and the Active Directory server. Specifically, the SSL property must be set to true. The Active Directory certificates must then be imported into the keystore of java running CA Identity Manager.

Before you attempt authentication with this module, the login name that is entered in the login screen must uniquely identify the same user in both the CA Identity Manager User Store and in Active Directory.

Specifically, CA Identity Manager searches for the user name that is entered in the login screen in the CA Identity Manager User Console by the attribute that is defined in the **Management Console, Environments, <Environment Name>, Advanced Settings, Authentication Properties, Authentication attribute to use** property. Typically, this attribute is defined as **%USER_ID%** or **%LOGIN_ID%**. Deployments can use some other attribute that uniquely identifies the user. The search filter property of the Active Directory Authentication module must define an attribute whose value can uniquely identify the Active Directory user. We recommend either **sAMAccount** or **userPrincipalName** .

Defining a configuration or entering a login ID value that fails to find both the CA Identity Manager user and the Active Directory results in an authentication failure.

The following table shows some common scenarios and the associated required configurations.

CA Identity Manager Configuration		**User Data**	
Authentication attribute	AD Authentication provider filter	CA Identity Manager User	Active Directory User
%USER_ID%	sAMAccountName=%s	userId=smithjo01	sAMAccountName = smithjo01
%LOGIN _ID%	sAMAccountName=%s	loginId=smithjo01	sAMAccountName = smithjo01

%LOGIN _ID%	userPrincipalName=%s	loginId=john.smith@mycompany.com (http://mycompany.com)	userPrincipalName = john.smith@mycompany.com (mailto:john.smith@mycompany.com)
%EMAIL%	userPrincipalName=%s	email= john.smith@mycompany.com (mailto:john.smith@mycompany.com)	userPrincipalName = john.smith@mycompany.com (mailto:john.smith@mycompany.com)

Use the following procedure to use the Active Directory authentication module class.

Follow these steps:

1. In the **Management Console**, select **Environments**, <environment_name>, and then click **Advanced Settings**.

2. In the **Authentication Properties** section, change the default value of **Authentication provider module class name** to

   com.netegrity.webapp.authentication.ad.ActiveDirectoryAuthenticationModule (http://com.netegrity.webapp.authentication.ad).

3. Locate **auth_settings.properties** file available at:
   <App_Server_DIR>/iam_im.ear/config/

4. Set the following properties in **auth_settings.properties** file:

   - **SERVERS**: Specifies the IP address of the Active Directory Servers. Use the following format (no spaces):
     IP1, IP2
     For example: 192.168.152.152,192.168.154.127

   - **ADMINPWD**: Specifies the Administrator Password for Active Directory. Enter and then confirm this password. This value is mandatory.

   - **BASEDN:** Specifies the Base DN for the User Search in Active Directory**.** This property is mandatory. For example: cn=Users,ca=companyX,dc=com

   - **ADMINDN**: Specifies the DN of the Administrator ID used to connect to Active Directory. This property is mandatory. For example: cn=Administrator,cn=Users,dc=companyX,dc=com

   - **SSL**: Determines whether to use SSL. Values are TRUE or FALSE.

   - **SEARCHFILTER**: Specifies a valid LDAP search filter with a variable substitution for an AD User. "%s" must be part of the filter, as it is replaced with the user name in authentication. This property is required. For example, to define a filter when using the default Active Directory User Schema, enter SEARCHFILTER=sAMAccountName=%s

Note: When using a custom Active Directory User schema, the objectCategory and ObjectClass filters clauses must be defined in the filter and match the LDAP object classes of the custom schema. For example, enter: SEARCHFILTER=(& (objectCategory=person)(objectClass=CompanyXUser)(sAMAccountName=%s))

5. Save the **auth_settings.properties** file and restart CA Identity Manager service.

Encrypting ADMINPWD and KEYSTOREPWD

You can simultaneously encrypt both the **ADMINPWD** (Administrator Password) and the **KEYSTOREPWD** (Key store password) using the **com.netegrity.webapp.authentication.ad.EncTicket** utility. This utility encrypts both **ADMINPWD** and **KEYSTOREPWD**.

This utility is packaged in the following location:

```
<IAM_IM.EAR>/user_console.war/WEB-INF/lib/user_console.jar
```

This utility is formatted as follows:

```
java -cp <IAM_IM.EAR>/user_console.war/WEB-INF/lib /user_console.jar com.netegrity.
webapp.authentication.ad.EncTicket <password>
```

For example:

```
C:\>java -cp C:/jboss/standalone/deployments/iam_im.ear/user_console.war/WEB-INF/lib
/user_console.jar com.netegrity.webapp.authentication.ad.EncTicket ChangeIt
```