HP Operations Manager

Authenticating Administration UI Users Using PAM or LDAP

Software Version: 9.10

for the UNIX and Linux operating systems

Manufacturing Part Number: None

January, 2014

Legal Notices

Warranty.

Restricted Rights Legend.

Copyright Notices.

Trademark Notices.

Conventions

The following typographical conventions are used in this manual:

Table 1 **Typographical Conventions**

Font	Meaning	Example
Italic	Book titles and manual page names	For more information, see the *HPOM Administrator's Reference* and the *opc(1m)* manual page.
	Emphasis	You *must* follow these steps.
	Variable that you must supply when entering a command (in angle brackets)	At the prompt, enter **`rlogin`** ***`<username>`***.
	Parameters to a function	The *`oper_name`* parameter returns an integer response.
`Computer`	Text and other items on the computer screen	The following system message displays: `Are you sure you want to remove current group?`
	Command names	Use the `grep` command...
	Function names	Use the `opc_connect()` function to connect...
	File and directory names	Edit the `itooprc` file... `/opt/OV/bin/OpC/`
	Process names	Check to see if `opcmona` is running.
`Computer Bold`	Text that you enter	At the prompt, enter **`ls -l`**.

Table 1 **Typographical Conventions (Continued)**

Font	Meaning	Example
Keycap	Keyboard keys	Press **Return**.
	Menu name followed by a colon (:) means that you select the menu, and then the item. When the item is followed by an arrow (->), a cascading menu follows.	From the menu bar, select **Actions: Filtering -> All Active Messages**.
	Buttons in the user interface	Click **OK**.

In This Document

This document describes how to authenticate Administration UI users using PAM or LDAP. Authentication of Administration UI users occurs inside the Administration UI WebApp server part to which the user's web browser connects.

IMPORTANT When setting up a new Administration UI user, make sure that the account exists in both Administration UI and the external authentication system. In addition, the Administration UI user must be a member of at least one Administration UI group that has at least one Administration UI user role assigned.

For detailed information, see the following sections:

❑ "PAM Authentication" on page 8

❑ "LDAP Authentication" on page 12

1 Authenticating Administration UI Users Using PAM or LDAP

PAM Authentication

To authenticate Administration UI users using PAM, no extra software is needed because Administration UI already includes the JPam open-source module. For details about JPam, see the following URL:

http://jpam.sourceforge.net

NOTE PAM is an interface linking software that provides authentication services such as LDAP, Kerberos, and UNIX passwd to user applications such as Administration UI. Therefore, software modules that implement the actual authentication service may be required.

To configure PAM authentication, follow these steps:

1. Decide which authentication service to use. If needed, install required software modules and configure them.

IMPORTANT It is highly recommended that you perform a stand-alone test of the authentication service (that is, outside the Administration UI context).

2. Configure all Administration UI user accounts in the authentication service.
3. Set up PAM authentication on the HP Operations management server.

   For details, see the *HPOM Administrator's Reference*.
4. Configure PAM to send Administration UI authentication requests to the desired authentication service (the PAM service name is `midas`).

NOTE PAM configuration is platform dependent. For troubleshooting, contact your system administrator.

For example, to use UNIX password authentication, perform the following:

- *On HP-UX:*

  Edit the `/etc/pam.conf` file for the `midas` module by adding the following lines:

  ```
  midas auth required \
  /usr/lib/security/hpux32/libpam_unix.so.1
  ```

  ```
  midas account required \
  /usr/lib/security/hpux32/libpam_unix.so.1
  ```

- *On Solaris:*

  Edit the `/etc/pam.conf` file for the `midas` module by adding the following lines:

  ```
  midas auth requisite pam_authtok_get.so.1

  midas auth required pam_unix_auth.so.1

  midas account required pam_unix_account.so.1
  ```

- *On RHEL:*

  Create the `/etc/pam.d/midas` PAM module, and then edit the `/etc/pam.d/midas` file by adding the following lines:

  ```
  auth sufficient pam_unix.so nullok try_first_pass

  auth required pam_deny.so

  account required pam_unix.so

  account required pam_permit.so
  ```

5. Activate the external authentication service in the `auth.properties` file by following these steps:

   a. Open the `auth.properties` file with the vi editor by running the following command:

   ```
   vi /opt/OV/OMU/adminUI/conf/auth.properties
   ```

   b. Edit the `auth.properties` file so that it contains the following:

   ```
   # external configuration file for complex authentication
   setups
   usermodel-router.authResource=file:conf/auth.xml
   # eof
   ```

6. Switch Administration UI to PAM authentication by configuring the `auth.xml` file.

   The following is an example file:

   ```
   <?xml version="1.0" encoding="UTF-8"?>

   <!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
   "http://www.springframework.org/dtd/spring-beans.dtd">

   <beans>

     <bean id="targetServices" class="java.util.ArrayList">

       <constructor-arg>

   <list>

    <value>pam</value>

    <value>usermgmt</value>

   </list>

       </constructor-arg>

     </bean>

   </beans>
   ```

   Administrator UI tries to use the PAM server for logon. If this authentication fails, Administration UI tries standard “user management” authentication.

   If you want to set up only PAM authentication (that is, without standard “user management” authentication), make sure that `auth.xml` contains only the `pam` value:

   ```
   <list>

     <value>pam</value>

   </list>
   ```

7. Deploy the `midas-wapam-sa.zip` service assembly by running the following command:

   ```
   cp /opt/OV/OMU/adminUI/assemblies/midas-wapam-sa.zip \
   /opt/OV/OMU/adminUI/deploy
   ```

8. Restart the WebApp by running the following command:

   ```
   /opt/OV/OMU/adminUI/adminui restart
   ```

The following is a test example (on Linux):

```
# export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/OV/OMU/adminUI/lib/midas
# echo $LD_LIBRARY_PATH
# /opt/OV/OMU/adminUI/adminui restart
# /opt/OV/OMU/adminUI/jre/bin/java -cp
/opt/OV/OMU/adminUI/lib/cli/midas_cli.jar:/opt/OV/OMU/adminUI/work
/service-assemblies/midas-wapam/version_1/sus/servicemix-lwcontain
er/midas-pam/lib/jpam-1.1.jar:/opt/OV/OMU/adminUI/lib/commons-logg
ing-1.1.jar com/bes/itm/comp/usermgmt/TestPam opc_adm opc_pam
```

LDAP Authentication

To authenticate Administration UI users using LDAP, no extra software is needed because Administration UI already includes the Acegi Security System for Spring Project open-source component. For details about this component, see the following URL:

http://acegisecurity.org

NOTE Currently, only basic authentication of user accounts is supported. No additional LDAP features such as group membership can be used.

When configuring LDAP authentication, choose one of the following two methods:

❏ Configuring LDAP Authentication Without Active Directory

❏ Configuring LDAP Authentication Using Active Directory

TIP To check the configuration values of your LDAP authentication configuration, you can use either Active Directory Users and Computers or the Apache Directory Studio open-source application.

Configuring LDAP Authentication Without Active Directory

To configure LDAP authentication without Active Directory, follow these steps:

1. Add all LDAP users that you want to authenticate to Administration UI, and then set the corresponding user roles.
2. Configure the desired LDAP server in the `ldap.properties` file (`/opt/OV/OMU/adminUI/conf/ldap.properties`) by following these steps:
   a. Configure a URL pointing to the desired LDAP server.

For example:

```
# The LDAP URL
# Format: ldap://<host>:<port>/<base dn>
# Format: ldaps://<host>:<port>/<base dn>
ldap.url=ldap://astrid:389/dc=hp,dc=com
#ldap.url=ldaps://astrid:636/dc=hp,dc=com
```

For both unencrypted and encrypted access, use `ldap.url=ldap://`.

NOTE

Make sure that you update the URL and the LDAP port based on your LDAP settings, as well as check your distinguished name (DN).

This example is used for the following scenario:

```
<host> : astrid
<port> : 389
<base dn>:  dc=hp,dc=com
Full URL: ldap://astrid:389/dc=hp,dc=com
```

In this instance, `dc=hp,dc=com` is the DN of the LDAP node that is marked as the initial context for LDAP operations. All subsequent LDAP operations (for example, `ldapsearch`) are performed on the subtree of that node.

IMPORTANT

Because the LDAP configuration is environment specific, make sure that you consult your LDAP administrator during the configuration process.

b. Continue with entering the log-on credentials. For example:

```
# Manager DN for login
ldap.managerDn=cn=Administrator,dc=hp,dc=com
# Manager password
ldap.managerPassword=******
```

In this instance, the `ldap.ManagerDn` property is the DN of the entry that is used to perform the BIND (authenticate) operation required for other LDAP operations (for example, `Search`, for

Administration UI). Keep in mind that the value of `ldap.managerPassword` must correspond to the password assigned to this entry.

c. Make sure that the LDAP authentication mode is set to the default value (that is, `BIND_WITH_DN`).

NOTE The LDAP authentication mode can also be set to `USER_SEARCH`, but it is highly recommended to use the default value.

With the default mode, usually no further configuration changes are necessary, so you can leave everything else commented out as shown in the following example:

```
# The mode which is used for the authentication
# Allowed values are:
# BIND_WITH_DN:Use the authenticationDnPatterns for
identifying a user
# USER_SEARCH : Use the authenticationSearchBase and
# authenticationSearchFilter for identifying a user
ldap.authenticationMode=BIND_WITH_DN
```

d. Add patterns for searching the users:

```
ldap.authenticationDnPatterns=sn={0},ou=People
```

In this instance, multiple patterns can be added, but they must be separated by vertical bars (`|`). These patterns represent Relative Distinguished Names (RDNs) that are relative to a root node configured in the `ldap.url` property. During authentication, `{0}` is replaced with a supplied user name.

For example, if a user wants to log on with the `admin` user name, `ldapsearch` searches for an entry with the following DN (this search is based on the previously specified configuration settings):

```
sn=admin,ou=People,dc=hp,dc=com
```

e. Verify the certificate. There are two possible scenarios:

- The certificate originates from a proper third-party certification authority such as Verisign.

  In this case, no other change should be necessary.

- A secure encrypted URL string is used, but without a certificate from a proper third-party certification authority.

  In this case, it is necessary to import the certificate into the local Administration UI truststore by following these steps:

  i. Configure the path to the truststore file and the truststore password as shown in the following example:

  ```
  # The path to the truststore for trusted certificates
  for secure LDAP
  ldap.truststore=conf/servicemix/truststore.jks
  # The truststore password for secure LDAP
  ldap.trustPassword=password
  ```

  ii. Import the `.cer` format certificate by running the following command:

  ```
  <JRE_path>/bin/keytool -import \
  -alias ldapserver_a -keystore \
  /opt/OV/OMU/adminUI/conf/servicemix \
  /truststore_endpoint.jks -file \
  /tmp/ldap_server.cer
  ```

  In this instance, <*JRE_path*> can be `/opt/OV/OMU/adminUI/jre` or `/opt/OV/nonOV/jre/b` (depending on your Administration UI version).

  iii. Answer the following questions:

  ```
  Enter keystore password: *******
  [...]
  Trust this certificate? [no]: yes
  ```

  The default password for the Administration UI truststore is `password`.

3. Activate the external authentication service in the `auth.properties` file by following these steps:

   a. Open the `auth.properties` file with the vi editor by running the following command:

   ```
   vi /opt/OV/OMU/adminUI/conf/auth.properties
   ```

b. Edit the `auth.properties` file so that it contains the following:

```
# configuration properties for authentication and
authorization components
#auth-filter.enabled=false
usermodel-router.authResource=file:conf/auth.xml
# eof
```

4. Switch Administration UI to LDAP authentication by configuring the `auth.xml` file.

The following is an example file:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://

www.springframework.org/dtd/spring-beans.dtd">

<beans>

  <bean id="targetServices" class="java.util.ArrayList">

    <constructor-arg>

    <list>

      <value>ldap</value>

      <value>usermgmt</value>

    </list>

  </constructor-arg>

  </bean>

</beans>
```

Administrator UI tries to use the LDAP server for logon. If this authentication fails, Administration UI tries standard “user management” authentication.

If you want to set up only LDAP authentication (that is, without standard “user management” authentication), make sure that `auth.xml` contains only the `ldap` value:

```
<list>

  <value>ldap</value>

</list>
```

Independent of whether LDAP or LDAPS is used, the default value must be `ldap`.

5. Deploy the `midas-waldap-sa.zip` service assembly by running the following command:

   ```
   cp /opt/OV/OMU/adminUI/assemblies/midas-waldap-sa.zip \
   /opt/OV/OMU/adminUI/deploy
   ```

6. Restart the WebApp by running the following commands:

   ```
   /opt/OV/OMU/adminUI/adminui clean
   ```

   ```
   /opt/OV/OMU/adminUI/adminui start
   ```

Configuring LDAP Authentication Using Active Directory

To configure LDAP Authentication using Active Directory, follow these steps:

1. Add all LDAP users that you want to authenticate to Administration UI, and then set the corresponding user roles.
2. Configure the desired LDAP server in the `ldap.properties` file (`/opt/OV/OMU/adminUI/conf/ldap.properties`) by following these steps:

   a. Configure a URL pointing to the desired LDAP server.

   For example:

   ```
   # The LDAP URL
   # Format: ldap://<host>:<port>/<base dn>
   # Format: ldaps://<host>:<port>/<base dn>
   ldap.url=ldap://electron:389/DC=eledc08,DC=lan
   #ldap.url=ldaps://electron:389/DC=eledc08,DC=lan
   ```

   For both unencrypted and encrypted access, use `ldap.url=ldap://`.

NOTE Make sure that you update the URL and the Active Directory port based on your LDAP settings, as well as check your DN.

This example is used for the following scenario:

```
<host> : electron
<port> : 389
<base dn> : DC=eledc08,DC=lan
Full URL : ldap://electron:389/DC=eledc08,DC=lan
```

In this instance, `DC=eledc08,DC=lan` is the DN of the LDAP node that is marked as the initial context for LDAP operations. All subsequent LDAP operations (for example, `ldapsearch`) are performed on the subtree of that node.

IMPORTANT Because the LDAP configuration is environment specific, make sure that you consult your LDAP administrator during the configuration process.

b. Continue with entering the log-on credentials. For example:

```
# Manager DN for login
ldap.managerDn=CN=Administrator,DC=eledc08,DC=lan
# Manager password
ldap.managerPassword=******
```

In this instance, the `ldap.ManagerDn` property is the DN of the entry that is used to perform the BIND (authenticate) operation required for other LDAP operations (for example, `Search`, for Administration UI). Keep in mind that the value of `ldap.managerPassword` must correspond to the password assigned to this entry.

c. Set the LDAP authentication mode to `USER_SEARCH` and, depending on the Active Directory server configuration, define the log-on name field as shown in the following example:

```
# The mode which is used for the authentication
# Allowed values are:
# BIND_WITH_DN : Use the authenticationDnPatterns for
identifying a user
# USER_SEARCH : Use the authenticationSearchBase and
# authenticationSearchFilter for identifying a user
ldap.authenticationMode=USER_SEARCH
# The search base for searching users for authentication
# This property is used in combination with the
# ldap.authenticationSearchFilter
# and is used e.g. for a Active Directory search
ldap.authenticationSearchBase=CN=Users
```

```
# The filter for searching users for authentication
# This property is used in combination with the
ldap.authenticationSearchBase
# and is used e.g. for a Active Directory search
ldap.authenticationSearchFilter=(sAMAccountName={0})
```

IMPORTANT

The value for `ldap.authenticationSearchBase` can be set to `CN=USERS`. This property is used in combination with `base_dn` from `ldap.url` to denote the base node for `ldapsearch` during authentication.

For detailed information, contact your LDAP administrator.

d. Verify the certificate. There are two possible scenarios:

- The certificate originates from a proper third-party certification authority such as Verisign.

  In this case, no other change should be necessary.

- A secure encrypted URL string is used, but without a certificate from a proper third-party certification authority.

  In this case, it is necessary to import the certificate into the local Administration UI truststore by following these steps:

  i. Configure the path to the truststore file and the truststore password as shown in the following example:

  ```
  # The path to the truststore for trusted certificates
  for secure LDAP
  ldap.truststore=conf/servicemix/truststore.jks
  # The truststore password for secure LDAP
  ldap.trustPassword=password
  ```

  ii. Import the `.cer` format certificate by running the following command:

  ```
  <JRE_path>/bin/keytool -import \
  -alias ldapserver_a -keystore \
  /opt/OV/OMU/adminUI/conf/servicemix \
  /truststore_endpoint.jks -file \
  /tmp/ldap_server.cer
  ```

  In this instance, `<JRE_path>` can be `/opt/OV/OMU/adminUI/jre` or `/opt/OV/nonOV/jre/b` (depending on your Administration UI version).

iii. Answer the following questions:

```
Enter keystore password: *******
[...]
Trust this certificate? [no]: yes
```

The default password for the Administration UI truststore is `password`.

3. Activate the external authentication service in the `auth.properties` file by following these steps:

   a. Open the `auth.properties` file with the vi editor by running the following command:

   `vi /opt/OV/OMU/adminUI/conf/auth.properties`

   b. Edit the `auth.properties` file so that it contains the following:

   ```
   # configuration properties for authentication and
   authorization components
   #auth-filter.enabled=false
   usermodel-router.authResource=file:conf/auth.xml
   # eof
   ```

4. Switch Administration UI to LDAP authentication by configuring the `auth.xml` file.

   The following is an example file:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://

www.springframework.org/dtd/spring-beans.dtd">

<beans>

  <bean id="targetServices" class="java.util.ArrayList">

    <constructor-arg>

    <list>

      <value>ldap</value>

      <value>usermgmt</value>

    </list>

  </constructor-arg>

  </bean>

</beans>
```

Administrator UI tries to use the Active Directory server for logon. If this authentication fails, Administration UI tries standard “user management” authentication.

If you want to set up only Active Directory authentication (that is, without standard “user management” authentication), make sure that `auth.xml` contains only the `ldap` value:

```
<list>
  <value>ldap</value>
</list>
```

Independent of whether LDAP or LDAPS is used, the default value must be `ldap`.

5. Deploy the `midas-waldap-sa.zip` service assembly by running the following command:

   ```
   cp /opt/OV/OMU/adminUI/assemblies/midas-waldap-sa.zip \
   /opt/OV/OMU/adminUI/deploy
   ```

6. Restart the WebApp by running the following commands:

   ```
   /opt/OV/OMU/adminUI/adminui clean
   ```

   ```
   /opt/OV/OMU/adminUI/adminui start
   ```