Using the UCP Web Service

This chapter describes the environment that you must set up to use the User Change Password (UCP) web service and explains how you can use it.

The UCP web service allows you to authenticate an internal user and change the internal user password. You can use this web service interface to integrate ACS with your in-house portals and allow users in your organization to change their own passwords.

The UCP web service allows only the users in your organization to change their passwords. They can do so on the primary or secondary ACS servers.

The UCP web service compares the new password that you provide with the password policy that is configured in ACS for users. If the new password conforms to the defined criteria, your new password takes effect. After your password is changed on the primary ACS server, ACS replicates it to all the secondary ACS servers.

The Monitoring and Report Viewer provides a User_Change_Password_Audit report that is available under the ACS Instance catalog. You can generate this report to track all changes made to user passwords in the internal database, including the changes made through the UCP web service. You can use this report to monitor usage and failed authentications.

Now, you can download the UCP.war file from ACS 5.8.1 and use it in the JBoss 5.1.0.GA application with jdk6.

Enabling the Web Interface on ACS CLI

You must enable the web interface on ACS before you can use the UCP web service. To enable the web interface on ACS, from the ACS CLI, enter:

```
acs config-web-interface ucp enable
```

For more information on the **acs config-web-interface** command, see *CLI Reference Guide for Cisco Secure Access Control System 5.8.1*.

Viewing the Status of the Web Interface from ACS CLI

To view the status of the web interface, from the ACS CLI, enter:

```
show acs-config-web-interface
```

For more information on the **acs config-web-interface** command, see *CLI Reference Guide for Cisco Secure Access Control System 5.8.1*.

This following sections describe how to use the UCP web service:

Understanding the Methods in the UCP Web Service

The UCP web service comprises the following method:

User Change Password

The User Change Password method authenticates a user against an internal database and changes the user password.

Input Parameters

- Username
- Current password
- New password

Purpose

Use the **changeUserPassword** method for applications that require a single-step procedure to change the user password. Changing a user password is normally a two-step procedure. The first step is to authenticate the user and the second step is to change the user password.

The **changeUserPassword** method allows you to combine the two steps into one. A script or a single-page web application is an example of applications that require a single-step procedure to change the user password.

To change a password:

1. Connect to the UCP web application

   A login page appears.

2. Enter the username and password.

   The **authenticateUser** web service function is invoked. If your credentials match the data in the ACS internal store, your authentication succeeds.

   Note: The user authentication process does not perform any change and does not authorize you to perform any task. You use this process only to verify if the password is correct.

If authentication succeeds, the web service compares the new password against the password policy that is configured in ACS.

If your new password meets the defined criteria, the **changeUserPassword** web service function is invoked to change your password.

Output Parameters

The response from the User Change Password method could be one of the following:

- Operation Succeeded
- Operation Failed

Exceptions

This method displays an error if:

- The authentication fails because of an incorrect username or password.
- The user is disabled.
- The password change operation fails because the password does not conform to the password complexity rules defined in ACS.
- A web service connection error occurs, such as network disconnection or request timeout error.

- A system failure occurs, such as the database being down and unavailable.

Using the WSDL File

This section describes the WSDL file and the request and response schemas for the User Authentication and User Change Password methods. This section contains:

Downloading the WSDL File

To download the WSDL file from the ACS 5.8.1 web interface:

1. Log into the ACS 5.8.1 web interface.
2. Choose **System Administration > Downloads > User Change Password**.
3. Click **UCP WSDL** to view the UCP WSDL file.
4. Copy the WSDL file to your local hard drive.
5. Click **UCP web application example** to download a sample web application and save it to your local hard drive.

UCP WSDL File

The WSDL file is an XML document that describes the web services and the operations that the web services expose. The UCP WSDL is given below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--************************************************-->
<!-- Copyright (c) 2009 Cisco Systems, Inc.-->
<!-- All rights reserved.-->
<!--************************************************-->
<definitions name="changepass"
targetNamespace="http://www.cisco.com/changepass.service"
xmlns:tns="http://www.cisco.com/changepass.service"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:MIME="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:DIME="http://schemas.xmlsoap.org/ws/2002/04/dime/wsdl/"
xmlns:WSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/">

<WSDL:documentation>
Copyright (c) 2009 Cisco Systems, Inc.

ACS5.8.1 WSDL
Service Interface for change password

This WSDL document defines the publication API calls for
changing user
```

```
password.
</WSDL:documentation>


<xsd:types>
<xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.cisco.com/changepass.service">


<xsd:simpleType name="UserNameType">
<xsd:restriction base="string">
<xsd:minLength value="1" />
</xsd:restriction>
</xsd:simpleType>

<xsd:element name="usernameType" type="tns:UserNameType" />

<xsd:simpleType name="PasswordType">
<xsd:restriction base="string">
<xsd:minLength value="1" />
</xsd:restriction>
</xsd:simpleType>

<xsd:element name="passwordType" type="tns:PasswordType" />

<xsd:simpleType name="StatusCodeType">
<xsd:restriction base="string">
<xsd:enumeration value="success" />
<xsd:enumeration value="failure" />
</xsd:restriction>
</xsd:simpleType>

<xsd:element name="ResponseType">
<xsd:complexType>
<xsd:attribute name="status" type="tns:StatusCodeType" use="required" />
<xsd:sequence>
<xsd:element name="errorMessage" type="xsd:string" minOccurs="0"
    maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</xsd:types>

<message name="AuthUserRequest">
<part name="user_name" element="tns:usernameType" />
<part name="password" element="tns:passwordType" />
</message>

<message name="AuthUserResponse">
<part name="authUserResponse" element="tns:ResponseType" />
</message>

<message name="ChangeUserPassRequest">
<part name="user_name" element="tns:usernameType" />
<part name="old_password" element="tns:passwordType" />
<part name="new_password" element="tns:passwordType" />
</message>

<message name="ChangeUserPassResponse">
<part name="changeUserPassResponse" element="tns:ResponseType" />
</message>

<WSDL:portType name="ChangePassword">
<operation name="authenticateUser">
```

```
<input message="tns:AuthUserRequest" name="authUserRequest" />
<output message="tns:AuthUserResponse" name="authUserResponse" />
</operation>

<operation name="changeUserPass">
<input message="tns:ChangeUserPassRequest" name="changeUserPassRequest" />
<output message="tns:ChangeUserPassResponse" name="changeUserPassResponse" />
</operation>
</WSDL:portType>

<WSDL:binding name="changePassSoapBinding" type="tns:ChangePassword">
<SOAP:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
<!--
This is the SOAP binding for the Change Password publish operations.
-->

<WSDL:operation name="authenticateUser">
<SOAP:operation soapAction="" />
<input>
<SOAP:body use="literal" />
</input>
<output>
<SOAP:body use="literal" />
</output>
</WSDL:operation>

<WSDL:operation name="changeUserPass">
<SOAP:operation soapAction="" />
<input>
<SOAP:body use="literal" />
</input>
<output>
<SOAP:body use="literal" />
</output>
</WSDL:operation>
</WSDL:binding>

<WSDL:service name="changepassword">
<documentation>
ACS5.8.1 Programmatic Interface Service Definitions
</documentation>
<port name="changepassword" binding="tns:changePassSoapBinding">
<SOAP:address location="https://localhost:8080/PI/services/changepass/" />
</port>
</WSDL:service>

</definitions>
```

Request and Response Schemas

This section lists the request and response schemas of the User Authentication and User Change Password methods. This section contains the following schema:

User Authentication Request

```
<message name="AuthUserRequest">
<part name="user_name" element="changepass:usernameType" />
<part name="password" element="changepass:passwordType" />
</message>
```

User Authentication Response

```
<message name="AuthUserResponse">
 <part name="authUserResponse" element="changepass:ResponseType" />
</message>
```

User Change Password Request

```
<message name="ChangeUserPassRequest">
<part name="user_name" element="changepass:usernameType" />
<part name="current_password" element="changepass:passwordType" />
<part name="new_password" element="changepass:passwordType" />
</message>
```

User Change Password Response

```
<message name="ChangeUserPassResponse">
<part name="changeUserPassResponse" element="changepass:ResponseType" />
</message>
```

Using the Python Scripts

You can create custom web-based applications to enable users to change their own password for your enterprise. This section describes how you can run a sample application that is developed using Python and provides the sample client code.

The ACS web interface provides a downloadable package that consists of:

- Python SOAP libraries for Linux and Windows
- Python script
- ReadMe—Contains installation instructions

To download this package:

1. Log into the ACS 5.8.1 web interface.
2. Choose **System Administration > Downloads > Sample Python Scripts**.

   The Sample Python Scripts page appears.
3. Click **Python Script for Using the User Change Password Web Service**.
4. Save the .zip file to your local hard disk.

Note: After installing ACS 5.8 patch 4, you must run the modules packaged in the Downloads option in ACS on RH 7 which has Python 2.7.5 and OpenSSL 1.0.1e that supports TLSv1.2/1.1 to use the python scripts of UCP on the Linux machine.

Sample Client Code, page 7 shows a sample.zip file. This file contains a .war file. You have to deploy this .war file within a web server, such as Tomcat. This example allows your application to communicate with ACS through the UCP web service.

Note: The Cisco Technical Assistance Center (TAC) supports only the default Python Script. TAC does not offer any support for modified scripts.

Sample Client Code

```
from SOAPpy import SOAPProxy

# Get the ACS host / IP
host = raw_input('Please enter ACS host name or IP address:\n')
targetUrl = 'https://' + host + '/PI/services/UCP/'

server = SOAPProxy(targetUrl, 'UCP')

# Get the username
username = raw_input('Please enter user name:\n')

# Get the old password
oldPassword = raw_input('Please enter old password:\n')

# Get the new password
newPassword = raw_input('Please enter new password:\n')

# Call the changeUserPassword with the given input
ans = server.changeUserPass(username, oldPassword, newPassword)

# Password changing failed
if ans.status == 'failure':
print '\nFailure:'

# Print all failure reasons
for err in ans.errors:
print err
else:
# Password was changed successfully
print 'Success'
```

Note: You must have Python software to run this script.

Using UCP.war File

You can use UCP.war file and install it in a JBoss or Apache Tomcat Server. After installing the UCP.war, you can access the UCP web services from the server and change the user password.

Installing UCP.war in JBoss Server

To deploy UCP.war in the JBoss server, complete the following steps:

1. Ensure that the *JAVA_HOME* is set correctly. *JAVA_HOME* is the location where JDK is installed.
2. Download and extract the JBoss 5.1.0.GA from the link *http://www.jboss.org/jbossas/downloads/.*
3. Start the JBoss server. You need to navigate to *<JBOSS_HOME>\bin* and run the **run.bat** (for Windows) or **run.sh** (for Linux) commands from this location to start the JBoss server. Check the command prompt for the clean startup of JBoss server. Contact the JBoss support team if the JBoss server does not start properly.
4. Enter *http://<JBOSS_Installed_Server_IP/HOSTNAME>:< PORT_CONFIGURED>* in a browser to launch the JBoss server. If the JBoss server does not start properly, then you need to contact the JBoss support team.

5. Stop the JBoss server using **<JBOSS_HOME>/bin/shutdown.bat** or **shutdown.sh** command.
6. Login to Cisco Secure ACS and download UCP.war from the path **System Administration > Downloads > User Change Password**.
7. Place the UCP.war file in the location *<JBOSS_HOME>\server\default\deploy* of the JBoss server.
8. Start the JBoss server. You need to navigate to *<JBOSS_HOME>\bin* and run the **run.bat** (for windows) or **run.sh** (for Linux) commands from this location to start the JBoss server. You need to verify the JBoss server for a clean startup.
9. Ensure that if the UCP directory is present in the location *<JBOSS_HOME>\server\default\work\jboss.web\localhost*. If you cannot find the UCP directory, then you need to download the UCP.war again and repeat the steps 7 to 10.
10. In case if you find any errors or exception even after you deploy UCP.war for the second time, you need to collect the logs from the following location *<JBOSS_HOME>\server\default\log* for further analysis.