Chapter 1: How to Configure Certificate-Based Authentication

Introduction

Product: CA ControlMinder

Release: All

OS: All

This scenario describes how a system or a CA ControlMinder administrator configures the Enterprise Management Server and end user machine to support certificate-based log in to CA ControlMinder Enterprise Management.

This Knowledge Base Article constitutes a portion of the official CA product documentation for this CA product. This Knowledge Base Article is subject to the following notices (see page 10), terms and conditions.

How to Configure Certificate-Based Authentication

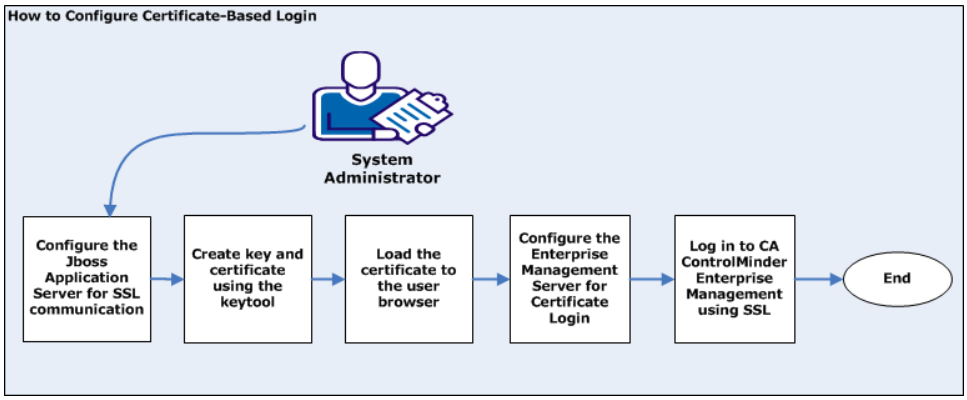
Certificate based authenticationBy default, communication to the Enterprise Management Server is not encrypted. To enable secured communication, you can configure the JBoss application server to use SSL to encrypt user credentials.

Further, you can configure the JBoss application server to support certificate-based login. In certificate-based login each user receives a unique certificate that is signed by a Certificate Authority (CA) to certify authenticity of the certificate.

When implemented on the client computer, certificate-based login does not require the user to supply their login credentials on each login. The user credentials are supplies by the certificate on each login.

Using certificates to authenticate users login enhances authentication capabilities of the Enterprise Management Server.

The following diagram illustrates how to configure certificate-based authentication on the Enterprise Management Server and client computer:



Follow these steps to configure certificate-based authentication:

1. Configure the JBoss application server for SSL communication (see page 3).
2. Create a key and certificate using the JDK keytool (see page 6).
3. Add a key to the client operating system. (see page 7)
4. Configure the Enterprise Management Server for certificate-based login (see page 8).
5. Log in to CA ControlMinder Enterprise Management (see page 10).

Configure the JBoss Application Server for SSL Communication

By default, JBoss is not installed with SSL support. This means that all communication between CA ControlMinder Enterprise Management and JBoss is not encrypted. You can configure JBoss to use SSL for secure communication.

Note: For more information about how to configure SSL for JBoss, refer to the JBoss product documentation.

Follow these steps:

This example shows you how to configure the JBoss application server to use SSL for secure communication.

Important! This procedure describes how to configure JBoss to use SSL for secure communication using JBoss version 4.2.3 and JDK version 1.5.0.

1. Stop JBoss if it is running.
2. Open a command-prompt window and navigate to the following directory, where JBoss_HOME indicates the directory where you installed JBoss:

   JBoss_HOME`/server/default/deploy/IdentityMinder.ear/custom/ppm/truststore`
3. Enter the following command to change the default ssl, keystore password:

   `keytool -storepasswd -new <password> -keystore ssl.keystore -storepass secret`

   -storepasswd

   Specifies to change the keystore password. The password must be at least six (6) characters long.

   -keystore

   Specifies the keystore name to add the certificate.

   -keystore

   Specifies the keystore name.

   -storepass

   Defines the password that is used to protect the keystore.

4. Enter the following command to create a key for the Enterprise Management Server:

   ```
   keytool -genkey -alias entm -keystore ssl.keystore -keyalg RSA
   ```

   -genkey

   Specifies that the command generates a key pair (public and private keys).

   -alias

   Defines the alias to add an entry to the keystore.

   -keyalg

   Specifies the algorithm to generate the key pair.

   The keytool utility starts.

5. Enter the password *secret*.

6. Complete the prompts as required and press enter to verify the parameters that you entered.

   The certificate is added to the keystore.

   Note: The keystore and key alias must use identical passwords.

7. Enter the following command to encrypt the keystore password to a file:

   ```
   java -cp C:/jboss-4.2.3.GA/server/default/lib/jbosssx.jar
   org.jboss.security.plugins.FilePassword welcometojboss 13
   <password> keystore.password
   ```

   Note: The Salt and IterationCount are the variables that define the strength of the encrypted password. In this example, "welcometojboss" is the salt and 13 is the iteration count.

8. Locate the file named server.xml in the following directory and open it for editing:

   ```
   JBossInstallDir\server\default\deploy\jboss-web.deployer
   ```

9. Locate the <Connector Port> tag in the following section:

   ```
   <!-- Define a SSL HTTP/1.1 Connector on port 8443
            This connector uses the JSSE configuration, when using
   APR, the
            connector should be using the OpenSSL style configuration
            described in the APR documentation -->
       <!--
       <Connector port="18443" protocol="HTTP/1.1"
   SSLEnabled="true"
                  maxThreads="150" scheme="https" secure="true"
                  clientAuth="false" sslProtocol="TLS" />
   ```

   Note: The connector port number corresponds to the JBoss HTTPS Port number that you specified during the prerequisite or CA ControlMinder Enterprise Management installation process.

10. Uncomment the "<!--" above the <Connector port> tag.

    You can now edit this tag.

11. Add the following properties to the <Connector port> tag:

    ```
    securityDomain="java:/jaas/encrypt-keystore-password"
    SSLImplementation="org.jboss.net.ssl.JBossImplementation"
    ```

12. Save and close the server.xml file.

13. Navigate to the following directory to locate the jboss-service.xml file:
    `JBOss_HOME/server/default/deploy/jboss-web.deployer/META-INF`

14. Add the following mbean between the <server> and </server> tags:

    ```
    <mbean code="org.jboss.security.plugins.JaasSecurityDomain"
    name="jboss.security:service=PBESecurityDomain">
          <constructor>
             <arg type="java.lang.String"
    value="encrypt-keystore-password"></arg>
          </constructor>
          <attribute
    name="KeyStoreURL">${jboss.server.home.dir}/deploy/IdentityMind
    er.ear/custom/ppm/truststore/ssl.keystore</attribute>
          <attribute
    name="KeyStorePass">{CLASS}org.jboss.security.plugins.FilePassw
    ord:${jboss.server.home.dir}/deploy/IdentityMinder.ear/custom/p
    pm/truststore/keystore.password</attribute>
    <attribute name="Salt">welcometojboss</attribute>
    <attribute name="IterationCount">13</attribute>
     </mbean>
    ```

    Note: In the preceding example, welcometojboss is the salt and 13 is the iteration count.

15. Save and close the jboss-service.xml.

    Note: After you complete this procedure, you can select to connect to JBoss, and CA ControlMinder Enterprise Management, in either SSL or non-SSL modes.

Create a Key and Certificate Using the Keytool

After you configure the JBoss application server for SSL communication, you create a key and certificate. The key enables user authentication on login with the credentials that the certificate contains.

Follow these steps:

1. On the Enterprise Management Server, open a command-prompt window and navigate to the following directory where *JDK* is the directory where you installed the Java Development Kit:

   JDK`/bin`

2. Run the following command:
   ```
   keytool -genkey -alias -keylag RSA -keystore client.jks -dname
   "CN=user,OU=my_org_unit,o=my_org,L=AA,ST=my_state,C=my_country"
   -storepass "password"
   ```

   -genkey

   Specifies that the command creates a key pair (public and private keys).

   -alias

   Defines the alias to use for adding an entry to the keystore.

   -keyalg RSA

   Specifiy to use the RSA algorithm to generate the key pair.

   -keystore *client.jks*

   Specify to add key *client* to the keystore using the jks format.

   -dname

   Specifies that the X.500 distinguished name for the generated certificate is "CN=user,OU=my_org_unit,o=my_org,L=AA,ST=my_state,C=my_country"

   -storepass "*password*"

   Specifies the password that protects the SSL keystore.

   The keytool utility generates the key.

3. Run the following command to export the certificate in a CERT format:

   ```
   keytool -export -file client.cert -keystore client.jks -storepass
   password -alias client
   ```

   -export

   Specifies to export the certificate.

   -file

   Specifies the certificate file to export.

   The keytool exports the certificate in a CERT file format.

4. Run the following command to create a key for a user add to the operating system keystore:

   ```
   keytool -importkeystore -srckeystore "client.jks" -destkeystore
   client.p12 -destkeystore PKCS12 -srcalias client
   ```

   -importkeystore

   Specifies to import an entire keystore into another keystore. All entries from the source keystore, including keys and certificates, are imported to the destination keystore.

   -srckeystore "client.jks"

   Specifies the source keystore from which to copy all keys and certificates.

   -destkeystore *client.p12*

   Specifies the destination keystore to copy all keys and certificated into.

   -srcalias *client*

   Specifies the source keystore alias.

   The keytool creates the key. Next you add the key to the client operating system keystore.

Add a Key to the Client Operating System

After you create a unique key for the user, the user adds the certificate to the operating system keystore.

Follow these steps:

1. Copy the client key from the Enterprise Management Server into a temporary directory on the user computer.
2. Double-click the key.

   The certificate import wizard opens.
3. Click Next and specify the file to import.
4. Click Next.
5. Enter the password that you have specified when you created the key, then click Next.
6. Click Next and then Finish to complete the wizard.

   A message appears indicating that the import process successfully completed. Next you configure the Enterprise Management Server to support certificate-based authentication.

Configure the Enterprise Management Server for Certificate-Based Authentication

After you add the key to the client computer, you configure the enterprise management Server to support certificate-based authentication. In certificate-based authentication the Enterprise Management Server validates the client key against a certificate that is stored on the server.

Note: To configure Enterprise Management for certificate-based authentication, you must enable the CA IdentityMinder Management Console. For more information about the CA IdentityMinder Management Console, see the *CA IdentityMinder Management Console online help.*

Follow these steps:

1. If running, stop the JBoss application server. Do *one* of the following:
   - Press Ctrl+C to close the application server window.
   - Stop the JBoss Application Server service from the Control Panel.
2. Open a command-prompt window and navigate to the following directory, where JDK is the directory where you installed the Java Development Kit:

   JDK`/bin`
3. To import the client certificate into the Enterprise Management Server keystore run the following command :

   ```
   keytool -import -file client.cert -keystore ssl.keystore
   -storepass secret -alias client
   ```

   -import

   Specifies to import a certificate into the keystore.

   -file *client.cert*

   Specifies the client certificate file to import into the keystore.

   -keystore *ssl.keystore*

   Specifies the name of the keystore.

   -storepass *secret*

   Specifies the keystore password.

4. Open the server.xml file. By default, the file is located in the following directory, where JBoss_HOME indicates the directory where you installed JBoss:

   JBoss_HOME/server/default/deploy/jboss-web.deployer

5. Locate the following entries to verify that you successfully imported the client key:

   ```
   truststoreFile="${jboss.server.home.dir}\deploy\IdentityMinder.
   ear\custom\ppm\truststore\ssl.keystore"
   truststorePass="secret"
   ```

6. Navigate to the following directory, where *ACServerInstallDir* indicates the directory where you installed the Enterprise Management Server:

   ACServerInstallDir\IAM Suite\Access Control\tools\samples\client certificate login\

7. Copy the ac_login_sso_cert.jsp file to the following directory:

   JBoss_HOME/server/default/deploy/IdentityMinde.ear/user_console.war/app/ac

8. If you have not done so before, enable the CA IdentityMinder Management Console.

   For more information about how to enable the CA IdentityMinder Management Console, refer to the *Implementation Guide*.

9. Using a web browser, enter the following URL to log in to the CA IdentityMinder Management Console:
   `http://enterprise_host:port/idmmanage`

   The CA IdentityMinder Management Console opens.

10. Select Environments, ac-env, User Console.

11. Locate the field Login Page to Use and specify the full path of the directory where the ac_login_ss_cert.jsp file is located. For example:

    `app/ac/ac_login_sso_cert.jsp`

12. Start the JBoss application server.

    You have successfully configured the Enterprise Management Server for certificate-based authentication. Now you can log in to CA ControlMinder Enterprise Management.

Log in to CA ControlMinder Enterprise Management

Once you configure the Enterprise Management Server for certificate-based authentication and add the client key to the operating system, you can login to CA ControlMinder Enterprise Management without providing your login credentials.

To login to CA ControlMinder Enterprise Management open a web browser and enter the following URL:
`https://enterprise_host:HTTPSport/iam/ac`

Note: You can also open CA ControlMinder Enterprise Management from a Windows computer where you installed it by clicking Start, Programs, CA, Access Control, Enterprise Management.

Example: Open CA ControlMinder Enterprise Management Using SSL

Enter the following URL into your web browser to open CA ControlMinder Enterprise Management using SSL from any computer on the network:
`https://appserver123:18443/iam/ac`

The URL suggests that CA ControlMinder Enterprise Management is installed on a host named appserver123 and uses the default CA ControlMinder Enterprise Management SSL port 18443.

Copyright