HCPAWRepTool

Certificate-Based Authentication

User’s Guide

Version 0.2

# Overview

The HCPAWRepTool is a java-based tool that performs HTTPS REST requests to the HCP-AW server’s reporting API to generate CSV files of requested user consumption information. The HCP-AW API requires authentication to collect this data. Older versions used only username and password information to authenticate; however, this is in sufficient for some customers. Therefore, this tool has been instrumented to authenticate with the HCP-AW server utilizing Certificate-Based Authentication (CBA). This document explains the different mechanisms to utilize the CBA mechanism.

To get started, two basic TLS/SSL concepts should be understood: trust and authentication. Trust is the mechanism to allow the client to identify and ensure it knows (trusts) the server it is communicating with. Authentication is how the server knows that the client is something that is allowed to make requests. To ensure the most secure connection, both trust and authentication should be used.

In this context, HCPAWRepTool is the client and HCP-AW server is the server.

# Establishing Trust

For establishing a TLS/SSL trust relationship during connection, during the initial interchange the server provides a certificate stating its identity. The client can decide to trust the certificate by ensuring that either that certificate or the certificate’s root certificate is trusted, or can just assume it is trusted (unsafe).

The default behavior of the HCPAWRepTool is to trust all server certificates (unsafe). If this is not the desired execution mode, the tool can be configured to enforce the verification of the server trust relationship by providing the following JVM run-time property. Add the following property definition to the java command line:

-Dhcpawreptool.enforceCertTrust=true

When using HCPAWRepTool in production, it is ***highly recommended*** that server certificate be validated for trust.

If the HCP-AW certificate is signed by a well-known commercial Certificate Authority (CA), there likely is nothing that needs to be done for HCPAWRepTool to trust the HCP-AW server, as the CA root certificate is likely in the default JVM trust store.

However, if it is not, one ***must*** acquire and configure an appropriate server or root certificate. If the HCP-AW certificate is self-signed, the HCP-AW certificate is all that is needed. If it is signed by a not-well-known CA, the CA root certificate must be acquired. How to acquire the self-signed or root CA certificate is beyond the scope of this document, but usually it can be obtained via various browser or OS tools.

To establish trust during connection, the certificate must be loaded either into the Java Run-time Environment (JRE) default trust store or loaded into a private trust store. The JRE default trust store path is rooted at JRE\_HOME under file path lib/security/cacerts. The file system location of JRE\_HOME will vary from OS and installation. Note: the default password for the cacerts file is changeit.

To load and trust the certificate into the default JRE trust store, locate the *cacerts* file and use the *keytool* utility provided with the JRE. The following is a *keytool* usage example:

*keytool -importcert -keystore ./cacerts -storepass changeit -file ~/Customers/DISA/CertAuth/HDSFLAB-CA-Root.cer -alias HDSFLAB-CA   
-trustcacerts -noprompt*

Alternatively, it is possible to create a private trust store with the required root certificate. The following is an example command:

keytool -importcert -keystore HCPRepToolTrust.jks -storepass changeit

-file HDSFLAB-CA-Root.cer -alias HDSFLAB-CA -trustcacerts -noprompt

To use this private trust store, the JVM must be told of the trust store location and password by executing the java command with definitions like the following:

-Djavax.net.ssl.trustStore=HCPRepToolTrust.jks

-Djavax.net.ssl.trustStorePassword=changeit

# Using Certificate-Based Authentication

Certificate-Based Authentication (CBA) is the mechanism to provide client credentials to the server to perform authentication against the server. CBA requires a CA issued certificate identifying the user and its associated private key. When generating the certificate, the user account must be included in the certificate’s User Principle Name field. The certificate and private key must be combined together in a Personal Format eXchange (PFX) file in PKCS12 format; additionally, it is highly recommended to password protect the file.

[NOTE: It is beyond the scope of this document to provide instructions on how this is accomplished, but there are plenty of examples that can be found on the internet.]

The HCPAWRepTool provides two mechanisms for providing credentials for CBA. One is from the command line, and the other is via execution parameters to the JVM java command.

From the HCPAWRepTool command line, provide the command arguments like the following:

--user-keystore ./mycert\_wkey.pfx --password mypassword

To provide the credentials via JVM run-time variables, provide java command qualifiers like the following:

-Djavax.net.ssl.keyStore=./mycert\_wkey.pfx

-Djavax.net.ssl.keyStorePassword=mypassword

NOTES:

* Providing username via the tool command parameters will cause the tool to NOT use Certificate-Based Authentication.
* Keystore information provided via the command line will take precedence over any JVM run-time variables.

## Windows Login Certificates

On Windows, there is a built-in certificate and key store that facilitates Single-Signon (SSO). The JRE has the ability to utilize this store by providing a specific indicator for the key and trust store names. With the HCPAWRepTool, this can be provided via one of two ways:

1. Provide HCPAWRepTool command line qualifier: *-k Windows-MY*
2. Supply JVM run-time variable: -Djavax.net.ssl.keyStore=Windows-MY