Run IDERA Dashboard over TLS (HTTPS)

The IDERA Dashboard Web Application service comes with TLS1.2 already set up. By default, TLS works with a self-signed certificate. This certificate can be used for encryption only and does not prove the identity of the server.

That default certificate is not signed by any well-known Certification Authority (CA), and is intended only for use in testing purposes. When a user attempts to open the TLS version of the IDERA Dashboard Web interface, a warning appears in the browser window.

If you decide to continue working with this self-signed certificate , you must perform several steps to "accept" the certificate before you can access the site. This step usually occurs only the first time you access the site. Then the self-signed certificate is stored in the browser database marked as trusted. This scenario is suitable for testing purposes.

If you want to obtain a CA-signed certificate , use the steps below to obtain a certificate signed by a well-known CA. The role of a CA is to verify that the IDERA Dashboard Web Application you are trying to access actually has the name you are trying to access it by, and that this server actually belongs to your organization.

Obtaining a CA-signed certificate

Certificates are issued by trusted third-party Certification Authorities (CAs). Many CAs simply verify the domain name and issue the certificate, whereas others (VeriSign, etc.) verify the existence of your business, the ownership of your domain name, and your authority to apply for the certificate, providing a higher standard of authentication.

Every browser comes with a pre-defined list of well-known CAs. You can find a sample list of CAs at http://www.dmoz.org/Computers/Security/Public_Key_Infrastructure/PKIX/Tools_and_Services/Third_Party_Certificate_Authorities/.

Along with the name of your organization and the name of your server, a CA-signed certificate contains the public key of the server. This public key is used by the browser to encrypt data sent to the server. There is a private key on the server. The server uses the private key to decrypt the data encrypted by the public key. The private key should be kept secure on the server to prevent unauthorized access.

For more information about public key cryptography, see http://en.wikipedia.org/wiki/Public-key_cryptography. To learn more about certificates and steps to buy a certificate, refer to a CA website such as:

- VeriSign
- Thawte
- CAcert
- GoDaddy

Generating a certificate request

Before the CA can issue you the certificate, you should generate private key and the certificate request and send it to the CA for signing. The certificate request and the private key should be generated using the `openssl` command unless otherwise instructed by the CA.

> While generating the private key and certificate request, replace the `openssl` command with the full path to binary, for example, `C:\Openssl\bin\openssl`.

Importing the certificate into the Trust-Store

The following steps show you how to install a SSL certificate purchased from a Certification Authority. Your SSL vendor may have different instructions, please check with them for proper certificate installation. The following examples refer to GoDaddy and VeriSign.

To enable a certificate, use the **Java keytool** - a key and certificate management utility. The keytool stores the keys and certificates in a so-called **keystore**.

It is assumed that you have both the private key file and certificate file in the PEM format and OpenSSL tool for Windows is installed into. It is also assumed that the private key file is called `wildcard.idera.com.key` and the certificate file is called `wildcard.idera.com.crt` and both are on disk C , in the root directory.

> You can download OpenSSL for Windows installation package from http://gnuwin32.sourceforge.net/packages/openssl.htm.

1. Start a Windows Command Prompt by clicking **Start > Command Prompt**. Alternatively, you can go to **Start > Run** and then type **cmd** without quotes and press <Enter>.
2. Use `C:` and then `cd\` commands to go to the root directory of the disk C, where the key and certificates are located.
3. Run the following commands to convert the key and the certificate from PEM to DER format.
   ```
   C:\OpenSSL\bin\openssl pkcs8 -topk8 -nocrypt -in wildcard.idera.com.key -inform PEM -out wildcard.idera.com.key.der -outform DER
   C:\OpenSSL\bin\openssl x509 -in wildcard.idera.com.crt -inform PEM -out wildcard.idera.com.crt.der -outform DER
   ```
4. Use the `cd` command to go to the directory where the keytool is located.
   ```
   cd "C:\Program Files\Idera\Dashboard\WebApplication\JRE\bin\"
   ```
5. Use Internet Explorer to download the ImportKey utility.

6. Point Internet Explorer to http://community.igniterealtime.org/servlet/JiveServlet/download/196707-4718/importkey.zip. Unzip the utility to `C:\Program Files\Idera\Dashboard\WebApplication\JRE\bin\` directory.
7. Run the following command. It will launch the ImportKey utility and create the keystore file (default name is keystore.ImportKey) in your home directory (in Windows 2008 it is usually `C:\Users\<your username>`). The private key and the certificate will be placed there.
   ```
   java ImportKey c:\wildcard.idera.com.key.der c:\wildcard.idera.com.crt.der
   ```

   > ⓘ The keystore and key passwords both must be set to **password**.

8. The following command allows you to set the password for your keystore file. The default password is **importkey**. Enter it when prompted, and then type the new password, which must be set to **password**.
   ```
   keytool -storepasswd -keystore c:\Users\Administrator\keystore.ImportKey
   ```
9. This command will allow you to set the password for the key file in the keystore. The default password is **importkey**. Enter it when prompted, and then type the new password, which must be set to **password**.
   ```
   keytool -keypasswd -alias importkey -ketstore c:\Users\Administrator\keystore.ImportKey
   ```
10. Use Internet Explorer to download the intermediate certificate chain for the Certification Authority (CA). For example, point Internet Explorer to https://certificates.godaddy.com/repository/sf_issuing.crt.
11. Save the intermediate certificate chain to the root directory of the disk C.
12. Import the received trusted certificate into your keystore file.
    ```
    keytool -import -alias intermed -file c:\sf_issuing.crt -keystore c:\Users\Administrator\keystore.ImportKey
    -trustcacerts
    ```

    > ⓘ Internet Explorer may change the file extension. If the command above does not work, try `sf_issuing.cer` instead of `sf_issuing.crt`.

13. Open Windows Explorer. Navigate to the directory `C:\Program Files\Idera\Dashboard\WebApplication\conf`.
14. Rename the file `keystore` to `keystore.old`. Then rename the file `C:\Users\<your username>\keystore.ImportKey` to `C:\Program Files\Idera\Dashboard\WebApplication\conf\keystore`.
15. Restart the IDERA Dashboard Web Application service.

keytool Options

- **alias**. All keystore entries are accessed via unique aliases. Aliases are case-insensitive. An alias is specified when you add an entity to the keystore using the `-import` command. Subsequent keytool commands must use this same alias to refer to the entity.
- **file**. Define absolute or relative path to your certificate file. If you define only file name, it means, that the file is located in the root directory.
- **keystore**. Each keytool command has a `-keystore` option for specifying the name and location of the persistent keystore file for the keystore managed by keytool. A keystore is created when you use `-import` command to add data to a keystore that does not already exist. If you do not specify a `-keystore` option, the default keystore is a file named `.keystore` in your home directory (as determined by the "user.home" system property). If that file does not already exist, it will be created.

Read more about Java keytool for Windows:
http://java.sun.com/javase/6/docs/technotes/tools/windows/keytool.html

IDERA Dashboard provides an integrated user experience for the IDERA products in your environment.

IDERA Website	Products	Purchase	Support	Community	About Us	Resources	Legal