Add/Import New PKCS#12 Certificate on the Cisco ESA GUI

Document ID: 117839

Contributed by Donny Lee, Cisco TAC Engineer.
Jun 25, 2014

Contents

Introduction
Prerequisites
Requirements
Problem
Workaround

Introduction

This document describes how to add/import new Public Key Cryptography Standards (PKCS) #12 certificates on the Cisco Email Security Appliance (ESA) GUI.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco ESA
- AsyncOS 7.1 and later

Problem

Since AsyncOS 7.1.0. and later, it is possible to manage/add certificates in the GUI of the email appliances. However, for this the new certificate, it has to be in PKCS#12 format, so this requirement adds some extra steps after receiving the Certificate Authority (CA) certificate.

Generating a PKCS#12 certificate also requires the Private Key Certificate. If you run the Certificate Signing Request (CSR) from Cisco ESA CLI command ***certconfig***, you will not receive the Private Key Certificate. The Private Key Certificate created in the GUI menu (***Mail Policies*** **>** ***Signing Keys***) will not be valid when you use it to generate a PKCS#12 certificate together with CA certificate.

Workaround

1. Install OpenSSL application if your workstation does not have it. The Windows version can be downloaded from here.

   Ensure that Visual C++ 2008 Redistributables is installed before the OpenSSL Win32.
2. Use a template to create a script to generate CSR and Private Key in here.

   The script will look like this:

```
   openssl req −new −newkey rsa:2048 −nodes −out test_example.csr −keyout test_example.key −subj
   "/C=AU/ST=NSW/L=Sydney/O=Cisco Systems/OU=IronPort/CN=test.example.com"
   ```

3. Copy and paste the script into OpenSSL window and press ***Enter***.

   ```
   C:\OpenSSL−Win32\bin>openssl req −new −newkey rsa:2048 −nodes −out test_example.csr
   −keyout
   test_example.key −subj "/C=AU/ST=NSW/L=Sydney/O=Cisco
   Systems/OU=IronPort/CN=test.example.com"
   ```

   Output:

   ```
   test_example.csr and test_example.key in the C:\OpenSSL−Win32\bin or in the
   'bin' folder where OpenSSL is installed
   test_example.csr = Certificate Signing Request
   example.key = private key
   ```
4. Use the .CSR file to request for the CA certificate.

5. Once you receive the CA certificate, save it as ***cacert.pem*** file. Rename private key file t***est_example.key*** to ***test_example.pem***. Now you can generate a PKCS#12 certificate using OpenSSL.

   Command:

   ```
   openssl pkcs12 −export −out cacert.p12 −in cacert.pem −inkey test_example.pem
   ```

   If the CA certificate and private key used are correct, OpenSSL prompts you to enter ***Export Password*** and confirm the password again. Otherwise, it advises you that the certificate and key that are used do not match and cannot proceed with the process.

   Input:

   ```
   cacert.pem = CA certificate
   test_example.pem = private key
   Export password: ironport
   ```

   Output:

   ```
   cacert.p12 (the PKCS#12 certificate)
   ```
6. Go to the IronPort GUI menu, ***Network*** > ***Certificate***.

   Select ***Add Certificate***.
   Select ***Import Certificate*** in the ***Add Certificate*** option.
   Select ***Choose*** and browse to the location of the PKCS#12 certificate generated in Step 5.
   Enter the same password that you used used when you generated the PKCS#12 certificate in the OpenSSL (in this case the password is ***ironport***).
   Select ***Next*** and the next screen will display the attributes details used for the certificate.
   Select ***Submit***.
   Select ***Commit changes***.

After these steps, the new certificate is added to the certificates list and can be assigned for use.