Trusted CA in iOS devices and iOS simulator

Rev 1 – February 11, 2013
By Daniel Cerutti – ADVTOOLS – http://www.advtools.ch

The list of trusted CA certificates consists of a set of built-in list plus the set of custom CA certificates added by the user. The list of custom CA certificates is maintained in a sqlite3 database file named TrustStore.sqlite3, which is located for the simulator at:

~/Library/Application Support/iPhone Simulator/VERSION//Library/Keychains/TrustStore.sqlite3

where VERSION is the IOS version (5.0, 5.1, 6.0, 6.1, …)

The TrustStore.sqlite3 is not available initially but will be created the first time a SSL connection is established from the simulator (for example by browsing to an https site from the simulator Safari browser).

TrustStore.sqlite3 consists of a single table named **tsettings** with the following structure

Name	Type	Description
sha1	blob	Certificate fingerprint (sha1 of the certificate data in binary DER form)
subj	blob	The full content of the certificate subject binary DER form (ASN1 encoding) with all PrintableString value in uppercase. The subject in a certificate consists of the following ASN1 structure: Sequence Set OID PrintableString (or other specific text type) Set OID PrintableString (or other specific text type) … The subj field contains this series of Set without the enclosing ASN1 sequence. Each PrintableString is converted to uppercase. Other text type values are left unmodified.

tset	blob	The following plist:
		
		`<?xml version="1.0" encoding="UTF-8"?>`
		`<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"`
		`"http://www.apple.com/DTDs/PropertyList-1.0.dtd">`
		`<plist version="1.0">`
		`<array/>`
		`</plist>`

data	blob	certificate in binary DER form

Before iOS 5.0, storing the a CA certificate in the data field and adding the sha1 was enough. Starting from iOS 5.0 it is required to correctly set all the fields, otherwise it will not be recognized by iOS, so the only convenient method to add a CA certificate to the iOS simulator was to add it to a physical device and the to copy the exact content of the device TrustStore.sqlite3 tsettings table (from a device backup) to the simulator.

Now the python script iosCertTrustManager.py, based on this information, allows to directly manage the list of custom CA certficates in the iOS simulator.