# Department of Computing

**CS381: Network Security**

**Class: BSCS-4**

# Lab 10: Digital Certificate

**CLO4: Compare, investigate and evaluate different security mechanisms to protect network resources**

**Date: November 20, 2016**

**Time: 9am -12 pm**

# Instructor: Ms. Haleemah Zia

# Lab 10: Digital Certificate

**Introduction**

In this lab lesson, the students will learn how to generate digital certificates using OpenSSL and keytool utilities. The certificate will then be configured with Tomcat and will use to create an https web page.

**Objectives**

The objective of this lab is:

1. Generate Self-Signed Digital Certificate
2. Configure Certificate with Tomcat for HTTPS

**Tools/Software Requirement**

OpenSSL utility, Tomcat, Java Keytool, Windows/Linux/OSX

**Description**

**Public Key Infrastructure (PKI)**

PKI can generate and issue digital certificates to the users. The OpenSSL is a command line tool, used for various cryptography functions such as:

1. Creation of RSA, DH and DSA key parameters
2. Creation of X.509 certificates, CSRs and CRLs (PKI)
3. Calculation of Message Digests
4. Encryption and Decryption with Ciphers
5. Handling of S/MIME signed or encrypted mail

**Instructions**

Complete the tasks below and insert the solution/answer in this document as directed below. You must show the execution of below tasks (by adding the snapshots), along with your required commands to get your work graded. You must also submit the completed Word document on LMS before the deadline. You can get help from the Internet, but copying is not allowed.

**Lab Tasks**

This lab has two main parts.

1. Generate keys and a Certificate Signing Request (CSR) for the public key certificate, and import the response from the Certification Authority (CA) using OpenSSL/Keytool utility.
2. Configure Tomcat with Certificate.

Follow all the steps and perform following tasks:

1. **Generate Digital Certificate using OpenSSL**

Configure and deploy a Public Key Infrastructure (PKI) using OpenSSL crypto library for creating certificates, CSRs and CRLs. Set up a multi-level Certificate Authority (CA). This multi-level CA should have one Root CA, one intermediate CA and a web-browser (end user).

1. **Download OpenSSL** from <http://gnuwin32.sourceforge.net/packages/openssl.htm> (its complete package, except source files)
2. **Create self-signed Certificates:** Follow the instructions given in the mentioned link to create self-signed certificate: <http://www.herongyang.com/crypto/openssl.html>

You need to perform all the steps mentioned in given links (red-boxed in below diagram). 

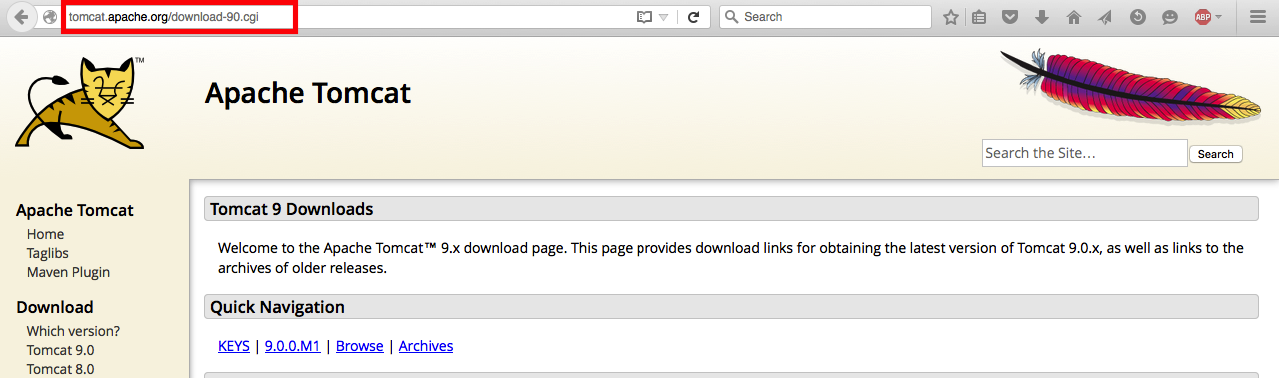
**Note:** For your convenience, learn the basic OpenSSL commands through the following link:

<http://users.dcc.uchile.cl/~pcamacho/tutorial/crypto/openssl/openssl_intro.html>

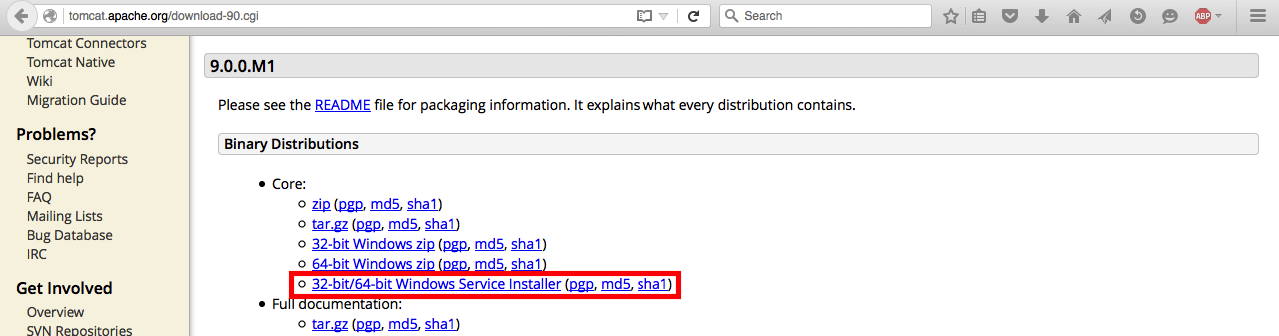
1. **Tomcat Configuration with Certificate**

**Installing Tomcat:**

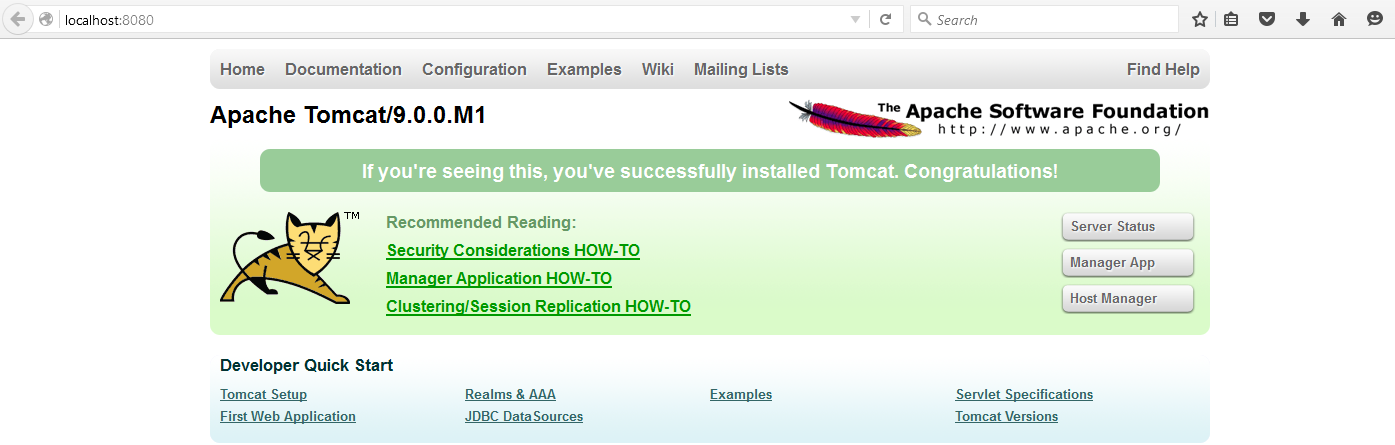
Download and install the apache tomcat setup from the apache website (<http://tomcat.apache.org/download-90.cgi>).



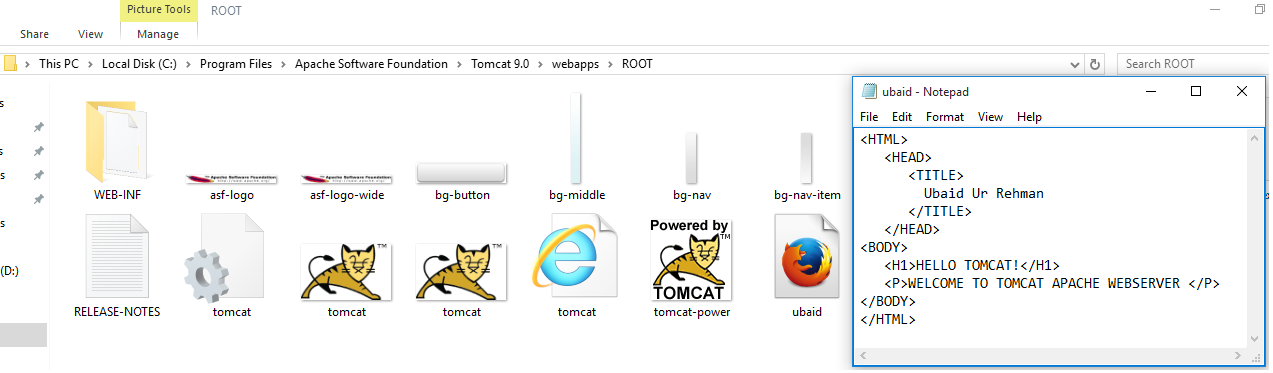
**Select** the **installer** as shown in figure below.



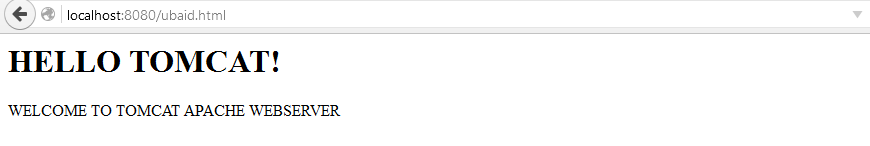
**Save** the installer. After the download is complete, double click the installer to **execute it**. Install the setup with default settings. To ensure that the server is started correctly, open your browser and type “**localhost:8080**” in the address bar.



Now create a simple webpage, for this you have to write HTML code. Either write your own code or get help from the Internet.

****

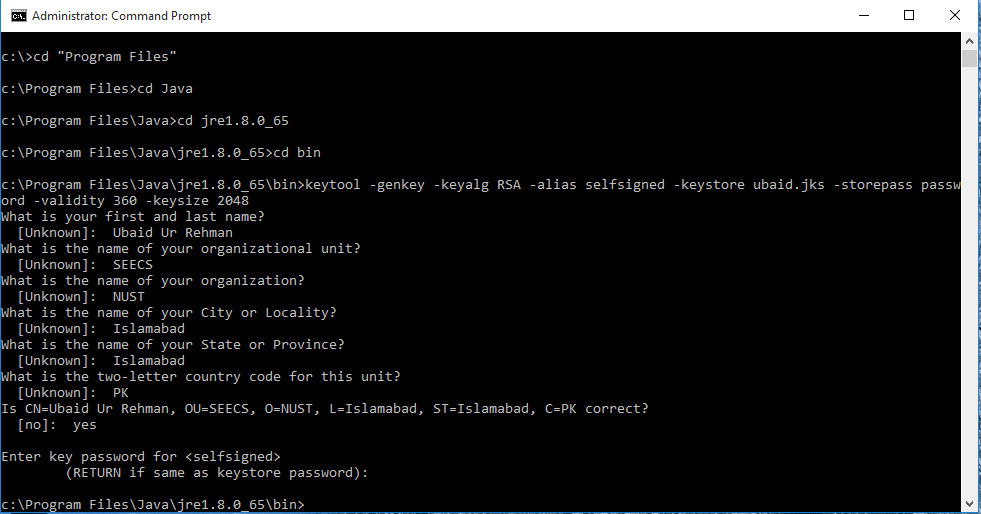
Now go to the web browser and type the name of the folder in which the “.html” is placed, in my case it is “C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\ROOT” folder (to run the page on local host).

****

Now, for converting the server to HTTPS, we need a certificate (created in section (a)).

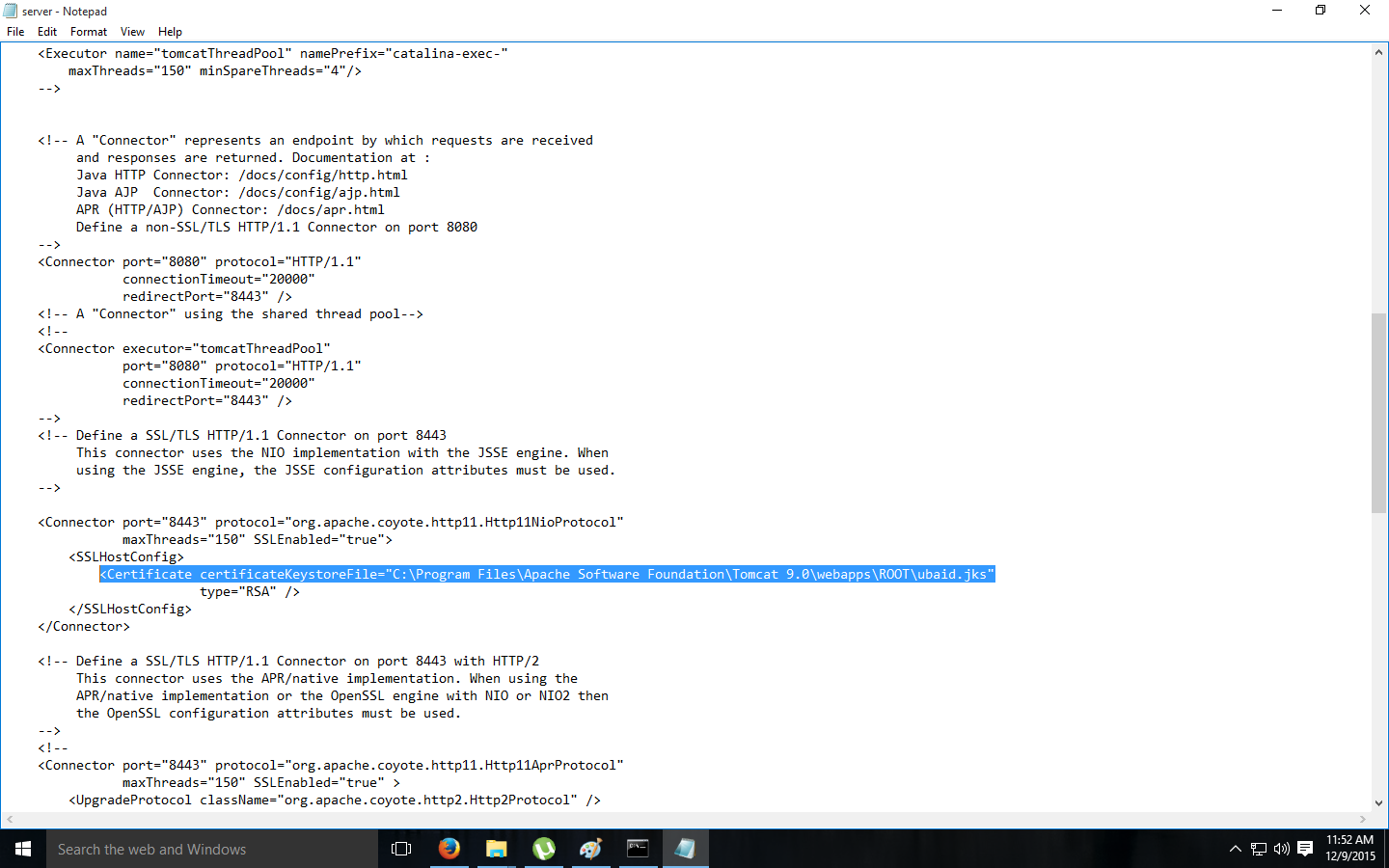
There are many ways and tools of generating Certificates, such as **OpenSSL**, **Java Keytool**, or using a GUI based software called **Portecle.** You can choose one as per your feasibility.

1. **Generated Digital Certificate using Java Keytool (2nd Method of creating certificate)**

****

1. **Configuration of Server**

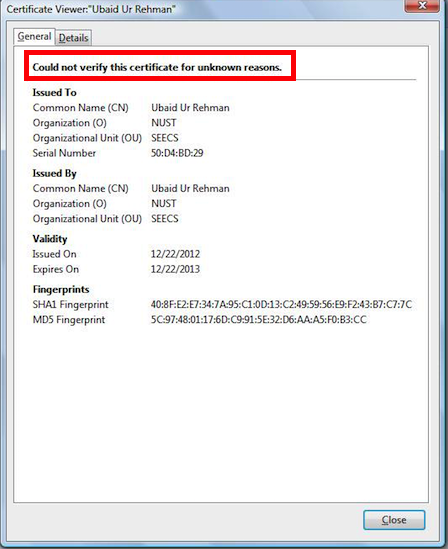
Changing the parameters of server.xml (C:\Progrtam Files\Apache Software Foundation\Tomcat 9.0\conf\server.xml).

****

Restart the tomcat server, and this time, go to **localhost:8443/ubaid.html**

|  |  |
| --- | --- |
| **Firefox Output** | **Google Chrome Output** |
|  |  |

**Warning shown, because of no ROOT certificate**



**However, your webpage has been shifted to HTTPS**

[2 Marks]

|  |
| --- |
| RSA Key Pair Generation |
| [\\add](file:///\\add) the snaps of the steps, which generate RSA Key Pair here |

[2 Marks]

|  |
| --- |
| Generating Self-Signed Certificate |
| [\\add](file:///\\add) the snaps of the steps, which generate self-signed certificate here. |

[2 Marks]

|  |
| --- |
| Display Self-Signed Certificate |
| [\\add](file:///\\add) the snaps of your generated self-signed certificate here |

[2 Marks]

|  |
| --- |
| Apache Tomcat |
| \\show the configuration of Apache Tomcat with port 8080 |

[2 Marks]

|  |
| --- |
| Configuration of Apache Tomcat |
| \\show the configuration of Apache Tomcat with port 8443. [1 Mark]  \\show the SSL enabled web page along with certificate (in browser). [1 Mark] |

**Deliverables**

Compile a single Word document by filling in the solution/answer part (as directed) along with the snapshots. Name your submission file as given below and submit this Word file on LMS before the deadline.

HadaiqAhmad **– 112807 – BSCS-4A NS gp1**

**Grade Criteria**

This lab is graded. Min Marks: 0. Max Marks: 10.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Minimum** | **Maximum** |
| Documentation with a clearly defined understanding of the lab task and approach | Fail | Pass |
| RSA Key Pair Generation | 0 | 2 |
| Generating Self-Signed Certificate | 0 | 2 |
| Display Self-Signed Certificate | 0 | 2 |
| Apache Tomcat | 0 | 2 |
| Configuration of Apache Tomcat | 0 | 2 |