Computer Security 2012

Project 1 - Digital Certificates

- This project should be done in groups of 1-2 people.
- You will not get detailed instructions of exactly how to do each step. Figuring out the exact steps is part of the project.
- The result of this project will be used in Project 2 so it is a good idea to document your work (to yourself).

1 Introduction

The goal of this project is to learn about certificates, what they are and how a certificate chain can be created in practice. In Project 2 you will *use* the certificates and the certificate chains. OpenSSL includes a library with several cryptographic functions. We will use functions related to certificates. You can find the OpenSSL documentation at [1]. We are mostly interested in the OpenSSL commands *req* and *x509*, but it might be useful to look also at other commands.

We will also use the program keytool, which is a tool included in the Java Development Kit. It is used for managing and creating key pairs and certificates, which are stored in keystores. However, keytool can not be used to sign other certificates. OpenSSL has to be used for this step. You can find the keytool documentation at [2].

The goal of this project is to create

1. A CA certificate that can be used to sign other certificates.
2. (Optional) A client certificate created with OpenSSL and signed with the CA certificate.
3. A client certificate created with keytool and signed with the CA certificate.
4. A keystore with a key pair for a client and a certificate chain with the CA certificate and the client certificate.

2 Project Instructions

Download and install OpenSSL and keytool. OpenSSL is included in most Linux distributions. If you use Windows you can find it at [3]. Read about certificates in the course book and then do the following steps.

1. Create a X.509 CA certificate using OpenSSL. Make sure to save the private key of the CA in a file.
2. (Optional) Create a *Certificate Signing Request* (CSR) for a client certificate using OpenSSL.
3. (Optional) Sign the CSR using the CA certificate.
4. Use keytool to create a user keypair that is stored in a keystore. Use your STIL identity as the *commonName* for the certificate. (If you are in a group of 2, just pick one STIL identity to use.)
5. Use keytool to create a CSR for the keys created in the previous step.

6. Use OpenSSL to sign the CSR with the CA created in the first step.
7. Import the certificate chain into your keystore.
8. Use keytool to verify that a certificate chain has been established. Your keystore should contain TWO entries, one CA certificate and one certificate chain consisting of the CA certificate and the signed certificate. The command "keytool -list -v ..." is appropriate.

To avoid writing many commands, it can be a good idea to write all steps in a batch-file or shell script. Then you can also reuse the batch file for the second project.

3 Getting Approved on Project 1

You get approved on Project 1 by submitting all files and corresponding passwords to Paul Stankovski (paul@eit.lth.se). He will first reply once to notify you that your report has been received. A second reply will, hopefully, let you know that your work has been approved. (If your keystore has three entries, do not bother sending your files. It will not get approved!)

References

[1] http://www.openssl.org/docs/apps/openssl.html

[2] http://download.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html

[3] http://www.slproweb.com/products/Win32OpenSSL.html