Team Name: Bay Area Rockers

Team Members: Shawn Chumbar, Dhruval Shah, Sajal Agarwal

CMPE 272: HW#7

**HW #7: Security**

In this assignment, we designed and built a PKI infrastructure which includes Root CA, Signing CA, and TLS Certificate. We used the following links for help regarding this assignment:

1. [Simple PKI Tutorial](https://pki-tutorial.readthedocs.io/en/latest/simple/)
2. [SSL/TLS Configuration How-To](https://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html)

**Section 1: Create Root CA**

The first step that we did was create the appropriate directories, create a database, create a CA request, and create a CA certificate. Please see below for the screenshots taken for this assignment followed by a description of the screenshot:

**A screenshot of a computer

Description automatically generated**

In the screenshot above, we create directories, create database, create CA request, and Create CA certificate. We tried to skip a line in between each set of commands. The overview for this section is as follows: The ca directory holds CA resources, the crl directory holds CRLs, and the certs directory holds user certificates. The database files must exist before the openssl ca command can be used. With the openssl ca command we issue a root CA certificate based on the CSR. The root certificate is self-signed and serves as the starting point for all trust relationships in the PKI. The openssl ca command takes its configuration from the [ca] section of the configuration file.

**Section 2: Create Signing CA**

In the screenshot below, the next thing that we do is create the signing CA. This includes the following steps:

1. Create directories.
2. Create database.
3. Create CA request.
4. Create CA certificate.

A screenshot of a computer

Description automatically generated

The ca directory holds CA resources, the crl directory holds CRLs, and the certs directory holds user certificates. We will use this layout for all CAs in this tutorial. With the openssl req -new command we create a private key and a CSR for the signing CA. You will be asked for a passphrase to protect the private key. The openssl req command takes its configuration from the [req] section of the configuration file. With the openssl ca command we issue a certificate based on the CSR. The command takes its configuration from the [ca] section of the configuration file. Note that it is the root CA that issues the signing CA certificate! Note also that we attach a different set of extensions.

**Section 3: Operate Signing CA**

In the two screenshots below, we perform the steps outlined in the **Operate Signing CA** section. This includes the following steps:

1. Create email request.
2. Create email certificate.
3. Create TLS Server Request
4. Create TLS Server Certificate
5. Create CRL

A screen shot of a computer

Description automatically generated

In the above screenshot, we perform the following commands: “create email request”, “create email certificate”, and “create TLS server request”. Please keep reading for a general summary of all these steps:

With the openssl req -new command we create the private key and CSR for an email-protection certificate. When prompted we entered the configuration specified in the PKI tutorial, which included the following DN components: DC=org, DC=simple, O=Simple Inc, CN=Fred Flintstone, emailAddress=fred@simple.org. We left other fields empty.

In the create email certificate step, we used the signing CA to issue the email-protection certificate. The certificate type is defined by the extensions we attach.

In the Create TLS Server Request step, we created the private key and CSR for a TLS-server certificate using another request config file. We entered these DN components: DC=org, DC=simple, O=Simple Inc, CN=www.simple.org.

The next screenshot shows the second set of commands we performed for section 3.

A screenshot of a computer

Description automatically generated

Please see below for a summary of the steps we did for the “Create TLS Server Certificate” and “Create CRL” steps.

We use the signing CA to issue the server certificate. The certificate type is defined by the extensions we attach. A copy of the certificate is saved in the certificate archive under the name ca/signing-ca/02.pem. The openssl ca -gencrl command creates a certificate revocation list (CRL). The CRL contains all revoked, not-yet-expired certificates from the CA database. A new CRL must be issued at regular intervals.

**Section 4: Output Formats**

This section encompasses the following steps:

1. Create DER Certificate
2. Create DER CRL
3. Create PKCS#7 Bundle
4. Create PKCS#12 Bundle
5. Create PEM Bundle

Please see below for the associated screenshots:

A computer screen with green text

Description automatically generated

Please continue reading for a summary of the screenshot:

All published certificates must be in DER format. All published CRLs must be in DER format. PKCS#7 is used to bundle two or more certificates. The format would also allow for CRLs, but they are not used in practice. PKCS#12 is used to bundle a certificate and its private key. Additional certificates may be added, typically the certificates comprising the chain up to the Root CA. PEM bundles are created by concatenating other PEM-formatted files. The most common forms are “cert chain”, “key + cert”, and “key + cert chain”. PEM bundles are supported by OpenSSL and most software based on it (e.g. Apache mod\_ssl and stunnel.)

**Section 5: View Results**

This section contains the following steps:

1. View request
2. View Certificate
3. View CRL
4. View PKCS#7 Bundle
5. View PKCS#12 Bundle

Please see below for the associated screenshots:

A screenshot of a computer

Description automatically generated

In the screenshot above, we perform the view request command. The openssl req command can be used to display the contents of CSR files. The -noout and -text options select a human-readable output format.

A screenshot of a computer

Description automatically generated

The screenshot above pictures us using the view certificate command. The openssl x509 command can be used to display the contents of certificate files.

A computer screen with green text

Description automatically generated

The screenshot above depicts us using the view CRL command. The openssl crl command can be used to view the contents of CRL files. Please note that we specify -inform der because we have already converted the CRL in step 4.2.

A screenshot of a computer screen

Description automatically generated

The picture above shows us using the View PKCS#7 Bundle command. The openssl pkcs12 command can be used to display the content of PKCS#7 bundles.

A screenshot of a computer screen

Description automatically generated

Lastly, the above screenshot shows us using the view PKCS#12 bundle. This command can be used to display the contents of PKCS#12 bundles.