CS 305 Module Five Coding Assignment: Certificate Generation

Student Name: Christian Busca

Course: CS 305 - Software Security

Instructor: Nicholas Lockwood

Date: February 7, 2025

Certificate Authorities

A **Certificate Authority (CA)** is a trusted entity responsible for issuing digital certificates. These certificates verify the authenticity of websites, users, or devices, ensuring secure communication over networks. CAs play a critical role in establishing trust in digital transactions by enabling encryption and authentication through **Secure Sockets Layer (SSL)** and **Transport Layer Security (TLS)** protocols.

Advantages of Using a Certificate Authority

- **Authentication:** Ensures the identity of a website or server is legitimate. This prevents attackers from impersonating a trusted entity.
- **Encryption:** Protects sensitive data by encrypting communication between clients and servers. This is crucial for online banking, e-commerce, and secure login systems.
- **Data Integrity:** Ensures transmitted data has not been altered during transit by verifying cryptographic signatures.
- **Trust Establishment:** Modern web browsers trust certificates issued by recognized CAs, preventing security warnings and allowing seamless access.

Why Use a Self-Signed Certificate?

A **self-signed certificate** is an alternative to CA-issued certificates, commonly used in development and testing environments. Unlike CA-issued certificates, which require validation and financial cost, self-signed certificates can be generated instantly without external verification. This allows developers to test **HTTPS connections** securely before deploying applications to production environments. However, since they are not verified by a trusted CA, users may encounter security warnings when accessing services secured by self-signed certificates.

Certificate Generation Process

To complete the assignment, a self-signed certificate was generated using the Java **Keytool** command-line utility. The following steps outline the process:

Step 1: Generate the Self-Signed Certificate

The following command was executed in the Linux terminal to generate a self-signed certificate and keystore:

keytool -genkey -keyalg RSA -alias selfsigned -keystore keystore.jks -storepass "SecurePass2024!" -validity 360 -keysize 2048

Entered Information:

- **First and Last Name:** CJ BUSCA
- **Organizational Unit:** Software Engineering
- **Organization Name:** SNHU
- **City or Locality:** Mustang
- **State or Province:** OK
- **Country Code:** US

Step 2: Export the Certificate

After generating the certificate, the following command was executed to export it to a .cer file:

keytool -export -alias selfsigned -storepass "SecurePass2024!" -file server.cer -keystore keystore.jks

Step 3: Verify and Print Certificate Details

The certificate details were verified using the following command:

keytool -printcert -file server.cer

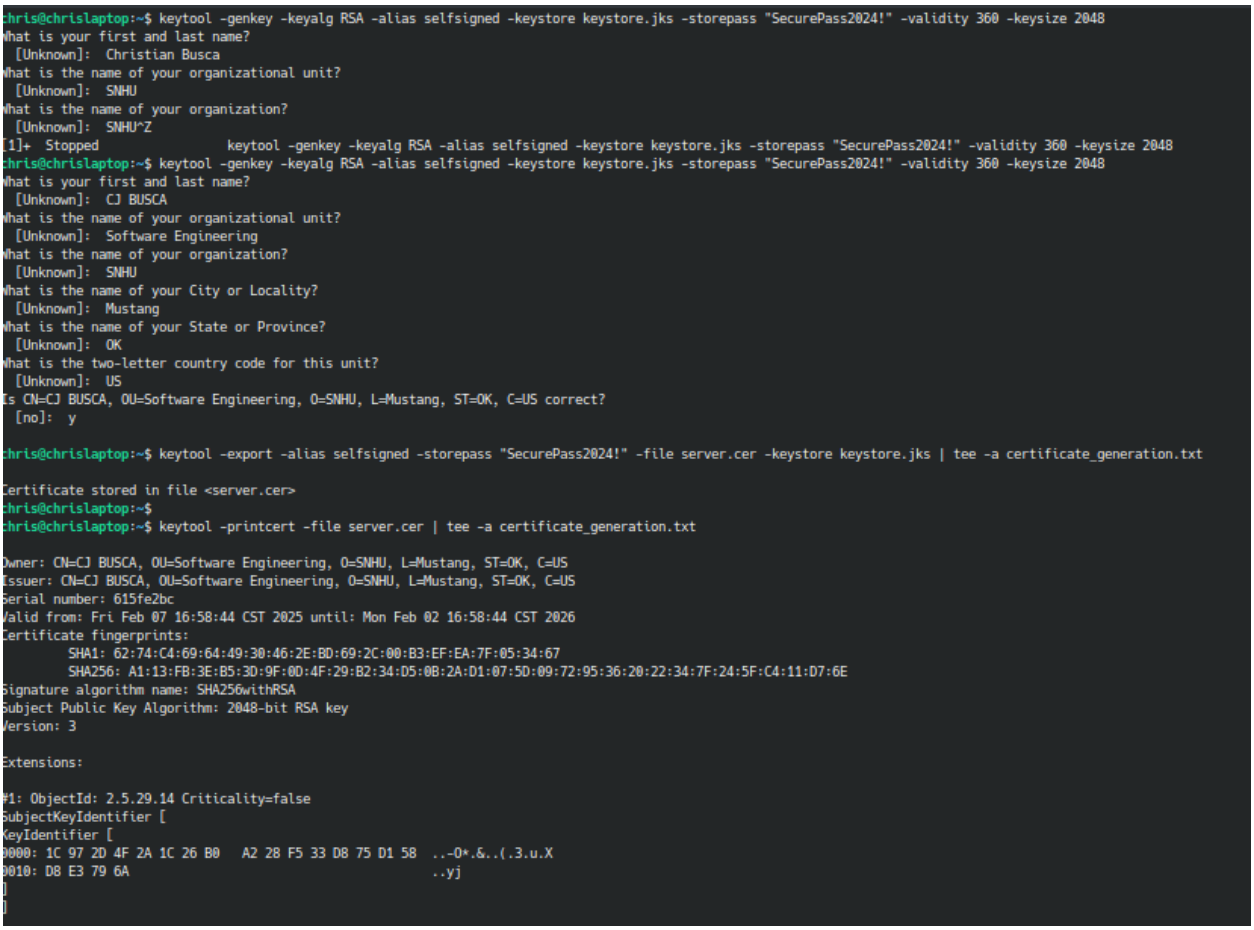
Output of the command:

Owner: CN=CJ BUSCA, OU=Software Engineering, O=SNHU, L=Mustang, ST=OK, C=US
Issuer: CN=CJ BUSCA, OU=Software Engineering, O=SNHU, L=Mustang, ST=OK, C=US
Serial number: 615fe2bc

Valid from: Fri Feb 07 16:58:44 CST 2025 until: Mon Feb 02 16:58:44 CST 2026
Certificate fingerprints:
SHA1: 62:74:C4:69:64:49:30:46:2E:BD:69:2C:00:B3:EF:EA:7F:05:34:67
SHA256:
A1:13:FB:3E:B5:3D:9F:0D:4F:29:B2:34:D5:0B:2A:D1:07:5D:09:72:95:36:20:22:34:7F:24:5F:C4:11:D7:6E
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Conclusion

In this assignment, a self-signed certificate was successfully created and verified using Java Keytool. The generated certificate provides **encryption, authentication, and integrity** for secure communications, simulating a real-world SSL/TLS certificate without the need for a third-party CA. Although self-signed certificates are useful for **development environments**, production systems should always use certificates issued by a trusted CA to avoid security warnings and establish full trust with end users.



References

Admin. (2020, April 21). What is a Certificate Authority? GlobalSign. Retrieved from https://www.globalsign.com/en-in/ssl-information-center/what-are-certification-authorities-trust-hierarchies

Detlefsen, A. (n.d.). Iron-clad Java. O'Reilly Online Learning. Retrieved from https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/

Techwalla. (n.d.). What are the Advantages & Disadvantages of a Digital Certificate? Retrieved from https://www.techwalla.com/articles/what-are-the-advantages-disadvantages-of-a-digital-certificate