SSL/TLS

OpenSSL

- OpenSSL is a very common SSL/TLS library
  - Written in C
  - Wrappers exist for many languages

- Can be used for many encryption needs
  - Generating keys
  - Signing certs
  - Validating certs

- We'll use OpenSSL in the command line to generate self-signed certificates

Self-Signed Certificate

```
openssl req -x509 -newkey rsa:4096 -keyout private.key -out cert.pem -days 365 -sha256 -nodes
```

- Once SSL is installed (Required on Windows) you can run commands in the command line

- This command will generate a self-signed certificate

Self-Signed Certificate

```
openssl req -x509 -newkey rsa:4096 -keyout private.key -out cert.pem -days 365 -sha256 -nodes
```

- This command has many options
  - You can adjust the options for your HW

- req
  - Request a signed certificate
- -x509
  - Use the x509 standard format for the certificate
- -newkey rsa:4096
  - Generate a new key for this cert using the RSA algorithm and a 4k key size

Self-Signed Certificate

```
openssl req -x509 -newkey rsa:4096 -keyout private.key -out cert.pem -days 365 -sha256 -nodes
```

- -keyout private.key
  - Save the private key in a file named "private.key"
- -out cert.pem
  - Save the public certificate in a file named "cert.pem"
- -days 365
  - This certificate will expire in 1 year
- -nodes
  - Do not require a password to use the private key

Installing the Certificate

- Now that we have a certificate, we need to use it in our server to enable TLS

- This will be different for each language/library
  - This is an excessive in reading technical documentation!
  - You will have to find and understand the documentation to use this certificate in your TCP setup
  - For some/most setups, this will be very simple and only require a few lines of code
    - Figuring out which lines of code and where to put them will be much more difficult!