Unit #2

- **Secure Sockets Layer (SSL) -** a protocol that encrypts, secures, and authenticates communications taking place on the Internet.
    - Cannot issue commands, unlike SSH.
    - Commonly used to secure client-server communications. *Eg. email, VoIP.*
        - The client has a public key. The server has both public and private keys.
    - Uses both symmetric and asymmetric encryption.
- **SSL certificate -** a digital document that validates a website's identity.
    - Essentially a set of public keys for a server.

Steps to an established SSL connection	
Certificate verification	● Client requests SSL certificate from the server. ● Client uses public key to verify the certificate.
TLS handshake	● Client generates a symmetric key. ● Client secures the symmetric key using the public key. ● Client sends the symmetric key to the server.
Key decryption	● Server receives the symmetric key from the client and decrypts it using the private key.
Data exchange	● The client and server can communicate securely by encrypting/decrypting messages with the same symmetric key.

- Clients are the entities that initiate a request for services or resources from a server. *Eg. web browsers, database clients, email clients, and mobile apps.*

Lab

- Folders are directories. Files are not directories.

Some CLI commands	
`ls`	Lists all the files and directories in the current directory.
`pwd`	Shows the directory currently in and the path taken from the root.
`sudo`	Lets you run other commands as the **root** user.
`cd [directory_name]`	Used to change directories.
`cd ..`	Used to move back up to the parent directory.
`tree`	Displays directory paths and files in each subdirectory.

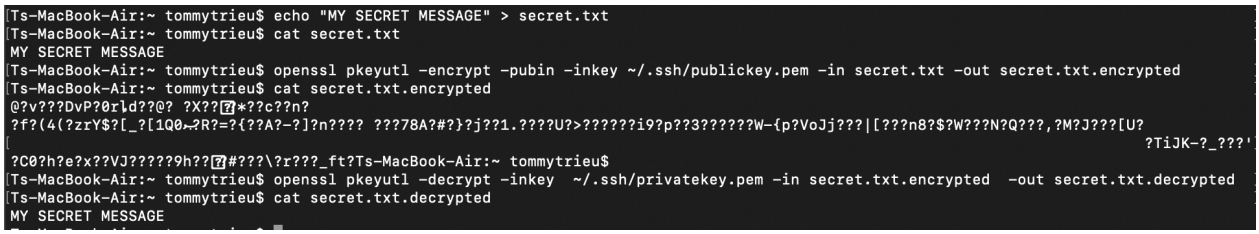
`touch [file_name]`	Used to add a new empty file into the current directory.
`wget`	Used to download content from web servers.
`cat [file]`	Outputs the content in a file.
`mkdir [folder_name]`	Creates a new folder in the current directory.
`cp [file_name] [duplicate_file]`	Makes a copy of a file.
`mv [file] [moved_file]`	Moves files and folders. It can also rename files and folders.
`rm [file]`	Deletes a file.
`rm -r [folder]`	Deletes a folder and everything in it.

Project

- **Secure Shell (SSH) -** a protocol that allows for remote access of a device such as a server over an insecure network.
    - It is used for managing networks, operating systems, and configurations.
- **SSH key -** authenticates the identity of a user or process that wants to access a remote system using SSH.

Command: `ssh-keygen -t ___ -b ___ -c "___"`
• `-t ___` specifies the type of key (the desired encryption algorithm). • `-b ___` specifies the number of bits. • `-c ___` is essentially a label/name for the key.

- **Passphrase -** similar to a password, it can be created and applied to the private SSH key for an extra layer of security.
- **Key fingerprint -** a unique identifier derived from a key. It is a way to verify the key's authenticity.

1. 

2.

```
codepath@lab000000:~/DemoProject$ echo -n "CYB101 Ubuntu Key" > ~/.ssh/git_allowed_signers && ssh-add -L >> ~/.ssh/git_allowed_signers
codepath@lab000000:~/DemoProject$ git config --global gpg.ssh.allowedSignersFile ~/.ssh/git_allowed_signers
codepath@lab000000:~/DemoProject$ git commit --allow-empty --message="Did the SSH signing work?"
[master (root-commit) e25a283] Did the SSH signing work?
codepath@lab000000:~/DemoProject$ git show --show-signature
commit e25a28305130304bd9ead7e480e16f51aaf4872b (HEAD -> master)
Good "git" signature with RSA key SHA256:EYCzEsD61VBcjBD0WE+VNQYRQMumWy3LKgixKZWbvao
/home/codepath/.ssh/git_allowed_signers:1: invalid key^M
sig_find_principals: sshsig_get_principal: key not found^M
No principal matched.
Author: Tommy Trieu <tommyt127@gmail.com>
Date:   Sat Sep 28 19:39:25 2024 +0000

    Did the SSH signing work?
```