# CS 405 Project Two Script Template

Complete this template by replacing the bracketed text with the relevant information.

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello, my name is Brett Barker. Today’s date is June the 19th, and today I will give you a presentation on software security. |
| **2** | What is Defense in Depth? Notice the model shown below.  This model gives us an overview of the defenses used within software development. This can help us maintain and produce secure software. |
| **3** | Here we will review what is known as a threat matrix.  This threats matrix table outlines levels of vulnerability. This can help measure the impacts of standards. |
| **4** | What are the basic principles to software security? Here below we can view the 10 principles, as well as some bonus principles.   * 1) Validate input data. * 2) Heed compiler warnings. * 3) Design and architect for security policies. * 4) Keep it simple * 5) Default deny. * 6) Adhere to the principle of least privilege. * 7) Sanitize data sent to other systems. * 8) Practice defense in depth. * 9) Use effective quality assurance techniques. * 10) Adopt a secure coding standard.   Bonus Principles:   * Define security requirements * Model threats. |
| **5** | After the principles we have standards. Here are the 10 basic standards.   * Do not depend on the order of evaluation for side effects. * Do not define a c style variadic functions. * Do not cast an out-of-range enumeration value. * Guarantee that indices and indicators are within valid range. * Do not attempt to create an std::string from a null pointer. * Do not access freed memory. * Close all files after they are no longer needed. * Do not abruptly terminate the program. * Honor replacement handler requirements. * Do not destroy a mutex while it is locked. |
| **6** | There are three encryption policies for secure coding. Let’s that a look at the three below.   * The three encryption policies are encryption at rest, encryption at flight, and encryption in use. * Encryption at rest is to prevent attackers to access data that is unencrypted. This is done by making the sure they data is encrypted when on a disk. This will then require access keys. * Encryption at flight is the process of encrypting data while it is being transported. This happens when data is unencrypted while resting but needs to be encrypted upon transportation. * Encryption is use is the practice of encrypting data that is at rest or in flight. This ensures sensitive data is never left unsecured. |
| **7** | We also have three triple a policies. Let’s view those three below.   * The three Triple-A consists of authentication, authorization, and accounting. * Authentication is used to make sure the user is who they claim to be. Some methods used are two factor authentication and multi-tier authentication. * Authorization is used to grant the users to certain permissions and access to what they are using. This could allow some user to delete file and other to just read/write. * Accounting is used to monitor the user of what they are doing with the access they have been given. This helps track what was done to databases and creates accountability. |
| **8** | Here we can view an example of unit testing. Unit testing throughout the development will help ensure code is working properly and securely. |
| **9** | Here we can view an automation summary for both pre-production and production. We will explain more on the next slide. |
| **10** | What we just saw can commonly be referred to as a DevSecOps pipeline.  A DevSecOps pipeline is the process of integrating security practices into your software development. When using this pipeline, you will be able to build and test your software more efficiently.  The layout of the DevSecOps pipeline in the previous slide is put together well. It has pre-production and the production laid out clearly. We can easily follow along with the steps without confusion. It is important to keep defense in depth in mind when starting a project. It is a great practice to test early and frequently. This will help you catch vulnerabilities early and will make it easier to find a solution. |
| **11** | What are some risk and benefits throughout the software development life cycle?  There will always be risks involved within the software development life cycle. Technology continues to progress and fast pace. It will be important to make updates and fixes as time progresses to fit the current technology we have at hand. When involving security, it is the best idea to always act now and do not wait. It is much easier to prevent a problem from happening than trying to fix it once it has been exploited. Security should be the highest measure for a developer, and it is a high priority for almost any user. If you decide to wait to implement your security measures, you are only doing a disservice to yourself and the users. There is almost no benefit gained from not implementing security from the start. |
| **12** | As technology changes the further time progresses, there will always be recommendation to be made.  I believe that the largest security gap in software security is not keeping up with current trends and technology. A large portion of security breaches come from outdated systems that haven’t been improved for current times. Not staying current on security measures is a great way to get exploited. My recommendation would be continuing to educate yourself on improvements that can be made to match the current technology. |
| **13** | In conclusion, with the standards and principles shown here today, we can conclude that we covered the topics to help us create and maintain secure and functioning code. It is important to apply the things explained here today to make sure you provide quality product to the client, while they can rely on the security of it. |
| **14** | Here are the references listed below.  *Sei cert C++ coding standard*. SEI CERT C++ Coding Standard - SEI CERT C++ Coding Standard - Confluence. (2020, May 29). Retrieved June 13, 2022, from https://wiki.sei.cmu.edu/confluence/pages/viewpage.action?pageId=88046682 |