# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Introduction-Jacob Perry |
| **2** | a type of cybersecurity in which several independent layers of security controls are used so that if one fails another will be operative. |
| **3** | Our matrix outlines the hierarchical importance of each of the identified potential issues. Priority taking precedence for fixing. |
| **4** | Validate Input Data Heed Compiler Warnings Architect and Design for Security Policies Keep It Simple Default Deny Adhere to the Principle of Least Privilege Sanitize Data Sent to Other Systems Practice Defense in Depth Use Effective Quality Assurance Techniques Adopt a Secure Coding Standard |
| **5** | Do not write syntactically ambiguous declarations  Do not dereference null pointers  Guarantee that storage for strings has sufficient space for character data and the null terminator  Prevent SQL injection – Ensure that when submitting user input for an SQL query, it’s sanitized to prevent escape statements from being passed.  Do not access freed memory  Assertions must be used to test for static parameters that will have a set specified parameter for testing, i.e. specific length/type/etc  Handle all exceptions  Ensure that unsigned integer operations do not wrap  Exclude user input from format strings  Detect errors when converting a string to a number |
| **6** | Utilizing a product such as IBM Guardium we can implement a TDE style encryption of our storage at rest. A TDE system encrypts and changes the structure of the files on the drive meaning that even if the data is compromised it still requires a key to access the newly encrypted structure.  While in flight, This is where the implementation of tools such as HTTPS, IMAPS, and SSH2 are critical to the protection of data during transit.  Finally we land at the in use phase of our encryption, this is the stage where our users or admins are working with data directly.  During this step we will implement a protocol such as Total Memory Encryption, introduced by Intel. This acts as a means of encrypting the data being stored in memory to ensure that it can’t be compromised. |
| **7** | Adhere to the principles of data validation, sanitation, and ensuring that no user input is leading to potential explorations.  When possible we will utilize authentication services such as Oauth2. Authorization will work simultaneously with a role system that will be implemented on the platform.  We will implement a thorough log system that will taken user actions throughout the platform/system and create a log for those actions. |
| **8** | Here is an output and source code for a series of unit tests.  These tests can and should be created for a project to ensure that the logic implemented within the code is protecting from identified threats. |
| **9** | Automation will be used for the enforcement of and compliance to the standards defined in this policy. Green Pace already has a well-established DevOps process and infrastructure. Define guidance on where and how to modify the existing DevOps process to automate enforcement of the standards in this policy. Use the DevSecOps diagram and provide an explanation using that diagram as context. |
| **10** | Implementing appropriate security processes and checks into the existing DevOps process can be achieved with minimal disruption. If we look back at our best practices and standards, in the pre-production phases or planning and designing, we can implement a majority of principles to inform how design decisions are made. Looking specifically at principle number 3, architect and design for security policies, by putting this at the forefront of security processes, we can ensure that the design stages of pre-production will have a focus on security. |
| **11** | * Waiting until a security breach has happened is later than a known risk should be resolved * Putting the platform at risk for legal repercussions * Later stage implementation can lead to further difficulties when integrating to an existing system * Higher levels of security from the start |
| **12** | * Future proofing the policy * Developing standardized tests that can bring to light new vulnerabilities * A devoted team that will work exclusively on updating the security protocols * Appropriate delegation of security concerns to the best team to implement a fix. |
| **13** | * Maintain a healthy code base * Keep it simple * Create regular checks for new vulnerabilities * No security policy is perfect, update at set intervals |
| **14** | [Insert text.] |

| During the authentication phase we will adhere to the principles of data validation, sanitation, and ensuring that no user input is leading to potential explorations. When possible we will utilize authentication services such as Oauth2 to provide a secure login portal that adheres to a larger services security standards. |
| --- |