# CS 405 Project Two Script Template

Fabian Rodriguez

<https://www.youtube.com/watch?v=Hei5G-fyqxk>

Project Two: Security Policy Presentation

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello, welcome to the Security Policy Presentation created for Green Pace |
| **2** | What is defense in depth?  A security policy is essential to all organizations, it helps to protect the systems and data to prevent security threats. By educating everyone about the best practices, we can create a robust system that can increase the security measures we could have in place. In the diagram below we can see the different layers of security that could be implemented when defense in depth is applied. |
| **3** | Threats Matrix, the show matrix, is what we use to assess risk, security and how important or frequent are the potential threats to our systems. We can see that in the upper left we have threats that are very frequent, in the upper right we have the highly relevant threats, while in the lower left we have the not so relevant ones. Finally, we have the threats that happen with low frequency over the lower right portion. |
| **4** | Let’s talk about the 10 principles. Starting with   * Validate Input Data – we must sanitize any data received from external sources * Heed Compiler Warnings - Warnings are there for a reason. * Architect and Desing for Security Policies - Security considerations must be incorporated into the architecture and design of any system * Keep It Simple - avoid unnecessary complexity * Default Deny - only be granting necessary permissions. * Adhere to the principle of Least Privilege – grant the minimum permissions required. * Sanitize Data Sent to Other Systems – we must send clean data * Practice Defense in Depth – layers of security help prevent threats * Use Effective Quality Assurance Techniques – Test, test, test * Adopt a Secure Coding Standard – adopt OWASP or CERT |
| **5** | Lets talk about Coding standards   * Range Verification – avoid index errors and out of bounds. * No Args C-Style functions – Always check the arguments passed. * Overflow Prevention – Ensure that assignment fit in the buffer * Explicit Code – The more explicit the better * Unsafe Memory Access – Prevent things as dereferencing pointers * Improper Assertions – Use the correct assertions * Robust Exception Handling – Catch exceptions properly * Variable Shadowing – must be avoided * Proper Resource Management – all files must be properly closed * Single Responsibility Code – Separate functionality |
| **6** | **Encryption in Rest** - The practice of encrypting the data when it is stored, for example sitting on a database. This applies depending on the sensitivity of the data and how the data is stored.  **Encryption at Flight** - The practice of securing the data when its being transmitted over networks or other channels. This is important to prevent data leaks even if the data is intercepted when traveling over the network or other channels.  **Encryption in Use** - Encrypting sensitive data while it is actively being processed or used by applications or systems. |
| **7** | **Authentication** - The process of verifying the identity of a person, system, or entities.  **Authorization** - After user are authenticated, authorization determines what actions or resources an authenticated entities can use or access.  **Accounting** - Keeps track of and logging user activities, system events, and access attempts |
| **8** | [Insert text.] |
| **9** | [Insert text.] |
| **10** | [Insert text.] |
| **11** | The DevSecOps pipeline is a method use to enforce code security by building and writing efficient code in a full circle.  There are multiple external tool we could use to enforce DevSecOps like monitoring services like log rocket, or simple a debugger. We could even add automated tools like automated testing or integration testing to ensure that we are meeting the standards and best practices. |
| **12** | There are always risk when dealing with software and systems. However, we can mitigate a lot of the risk by following the best practices and standard procedures. Implementing Triple A and Defense in Depth have a lot of benefits such as reducing the risk of security threats by a large amount. |
| **13** | Keeping dependencies up to date  Frequently scan the software for vulnerabilities  Encrypt data when necessary  Always practice the Principle of Least Privilege |
| **14** | With the standards outline in this presentation, and the guidelines provided in the security specification, we should be able to strongly implement and enforce good practices that would prevent security threats. However, security should be implemented at all levels and for any specification or standard to work we must have everyone on board and apply the best practices on all systems and software. |