# CS 405 Project Two Script

**Dan Taylor, 8/20/2023, Project Two: Security Policy Presentation**

[**https://youtu.be/foC9UkCeiK0**](https://youtu.be/foC9UkCeiK0)

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello everyone and welcome to the Green Pace security policy presentation. I am Dan Taylor, and I am here to take you through a basic overview of the policy as well as some of the details and benefits it will provide. Let’s get started. |
| **2** | The Green Pace Security Policy has been established to ensure that every layer of the organization is secure and held to a standard. The policy utilizes the Defense in Depth (DiD) strategy to ensure that multiple layers of security exist to mitigate the risk of any breach in the system. By implementing multiple layers of security, we provide multiple redundancies of protection if one or more of the layers should be breached or fail. These layers could mean the difference between a hacker gaining access to the system or not, and that can mean saving a lot of time, money, or reputational damage. |
| **3** | The threat matrix highlights coding standards and rules that should be followed to prevent the possibility of an exploitation or vulnerability. It displays each with a likelihood of occurrence, threat severity, remediation cost, priority, and level. We will dive into these and some basic principles that align with these ideas in the next few slides. |
| **4** | The ten principles on this slide are part of what the entire policy is based upon. The principles are 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, Validate Input Data, Heed Compiler Warnings, Architect and Design for Security Policies, Keep It Simple, and Default Deny. Some of these principles are self-explanatory, for instance keep it simple is saying just that. Some of the principles require a little explanation like Adhere to the principle of least privilege, it basically is saying not to give anyone access or permissions that they don't expressly need or are cleared to have. But generally speaking, they are a good base to set a policy on for information security in coding. |
| **5** | The same coding standards from the threat matrix are listed again here with a short description of what each of the standards is about. Instead of going into extreme detail or listing each standard, I will highlight a few of them. [Highlight several coding standards and briefly explain] |
| **6** | Encryption in rest describes when data is encrypted while it is being stored. The data may be accessible but will not be readable without a proper key. This encryption will help an organization to stay protected against attempts to steal data. Encryption at flight describes when data is encrypted while being transmitted. Data will be encrypted while being transferred to another storage location to protect sensitive data in the event it is intercepted by an outside entity or unauthorized party. The data will not be readable without the proper key. Encryption in use describes when data is encrypted while being used and allows certain access to certain users depending on authority level or necessity. Default Deny and the Principle of Least Privilege apply in this case as this helps to ensure data is accessed only by those who have an absolute necessity to access that data. |
| **7** | The Triple-A policies are as listed: Authentication, Authorization, and Accounting. Authentication is the process used to prove who a user is by, user ID, passwords, possibly higher-level security such as secure tokens, CAC/PIN and other hardware credentials. Authorization is once a user is authenticated and allowed access to a system, they are granted specific access to parts of that system. Authorized access to certain drives, folders, programs, or data allowed by the system administrators. The activity of all users on the system should be monitored and recorded after they are granted access through authentication and authorization. This explains Accounting. |
| **8** | Unit Testing is a method of ensuring code does not have specific vulnerabilities. The next few slides contain some basic examples of unit tests and the results. |
| **9** | DoesResizingIncreaseVector |
| **10** | Automated testing of code should be implemented as often and as early as possible. It can increase efficiency in the development process by avoiding major errors or adjustments later in the project. |
| **11** | Lightweight Directory Access Protocol (LDAP) can be used to query user information rapidly. LDAP controls the access and maintenance of static data such as usernames, passwords, email addresses, and other sensitive. Active directory (AD) is used to store sensitive information and is a secure way to control access to network resources. |
| **12** | The risk of waiting to implement security policies is much greater than the risk if implementing security policies early on. By waiting, you risk time, money, and reputational damage, and if the program is completed and ends up having many vulnerabilities it would be very costly. To complete an entire program and think of security at the end is very risky.  The solution is to begin implementing security from the very start. By doing this, there is a much higher level of security throughout the program, and there is a much lower risk of a vulnerability. |
| **13** | The recommendations that I make as a general rule moving forward are to encrypt everything, follow the triple-a framework, utilize unit test and external tools to test for vulnerabilities. |
| **14** | In conclusion, the security principles and standards that make up this policy are all important, but I hope you will walk away with the following at a minimum. Security standards should be implemented starting at the beginning of the development process during the planning phase.  DiD (Defense in Depth) should be used when developing any program or system, multiple layers of security mitigate the risk of a breach even in the event one security measure fails.  Keep processes simple and efficient to minimize errors and vulnerabilities.  Thank you! |