# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello everyone, my name is Sarah Snow, and this is my Security Policy Presentation for GreenPace. |
| **2** | This security policy employs a defense in depth strategy, using a multi-layered approach to security. This kind of approach ensures multiple hurdles for malicious users to overcome in order to reach any kind of sensitive data that Green Pace may store. A robust defense system is necessary not only to deter different attacks, but ultimately protects the company as a whole by upholding its reputation. |
| **3** | This threats matrix explains the kinds of threats that we may refer to throughoyt this presentation. Likely security risks are risks that have a high likelihood to occur because of gaps in the security coding standard, such as SQL Injection prevention. Priority threats are threats with high likelihood that pose the most detrimental threats to sensitive data and resources. SQL Injection would also fall under this category. Low priority threats, while still likely, pose less of a threat to sensitive data, but should still be addressed. Unlikely threats are threats with a low chance of happening, but their priority or severity are not determined by their likelihood. |
| **4** | This slide outlines the ten security principles we refer to when we are building a secure coding policy. The standards we address fall under one or more of these principles, as listed here. As you can see, validation, simplicity, practicing defense in depth, and adopting a secure coding standard are heavily weighted when designing a defense system. |
| **5** | These are the coding standards addressed in the secure coding policy. I have ordered them here in a list organized by balancing three different factors. The first factor I considered would the level of severity. As you can see, SQL injection would have the highest level of severity due to the risk of a data breach should the injection be successful. The second factor I addressed after severity was likelihood. While there may be some vulnerabilities that would be more likely than others, I placed severity ahead of likelihood to address the issues that would pose the biggest threat to the company if they were to happen. The third factor I considered while organizing my list was the cost of remediation. Keeping in mind the cost put some vulnerabilities that have similar severity and likelihood into order. Most companies would consider a vulnerability with a higher cost of remediation above another because should that vulnerability be exploited, the company’s bottom line would be more severly affected. |
| **6** | Encryption at Rest: Used to encrypt data during storage on devices such as databases, cloud servers, or hard drives. This prevents unauthorized access by using algorithms to encrypt entire disks. This applies whenever sensitive data is stored. It ensures that the data remains protected if the storage device is accessed without authorization.  Encryption in Flight: Used to secure data during transmission over a network, ensuring confidentiality and prevents interception. It is used to secure systems and communication between them such as servers or applications and databases. This is applied whenever data is transmitted over untrusted networks. It ensures the confidentiality of sensitive data. Use end-to-end protection, secure and updated libraries.  Encryption in Use: Used to encrypt data while it is being used by an application. The data stays encrypted using cryptographic libraries or APIs. Use identity management and the principle of least privilege. |
| **7** | Authentication: Verifies the identity of users accessing a system or application. Authentication usually uses passwords, biometrics, multi-factor authentication (MFA), and single sign-on (SSO). It also controls access to systems or data based on the identity of users. This applies whenever access control is required to protect the integrity sensitive data.  Authorization: Determines the permissions given to verified users. This employs the principle of least privilege, using control lists (ACLs), attribute-based access control (ABAC), and role-based access control (RBAC).  Accounting: Traces and records user activity involving a system. It ensures security by using audit trails, logging, etc. This helps detect malicious use of a system and applies whenever a user is accessing a system. |
| **8** | The unit tests I have written here are designed to handle SQL Injection attacks. There are three negative and one positive attack. There is a test that asserts valid input, a test for empty input and two tests for SQL Injection attempts. |
| **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. Automating the enforcement of the standards in the security policy should be integrated throughout the entire process and infrastructure.   + Access and Planning Phase: This is where project requirements should include security requirements. This includes secure coding practices such as data type validation and input sanitization. User stories and acceptance criteria should also consider security.   + Design and Build Phases: This is where security checks should be integrated. This is where static code analysis tools be implemented into the IDEs to provide immediate feedback regarding security issues.   + Verify and Test Phase: This should have automated security testing integrated into it. Static and dynamic analysis tools for runtime testing can identify potential vulnerabilities. Test cases that are security focused should be included. This is also where the code should be submitted for a security review before it is released to production.   + Transition and Health Check Phase: Controls should be integrated to securely configure the environment. Access controls and encryption should be implemented in this phase. Continuous monitoring and logging should be implemented in the monitor and detect phase. This can catch attacks in real time. Security information and event management (SIEM) tools can automate the detection of suspicious activity.   + Response Phase: This is where procedures should be implemented to ensure a timely and organized response to threats and events. |
| **10** | * Acting now to implement the security measures outlined in the policy can reduce the organization's vulnerabilities, protect sensitive data, as well as protect against potential breaches. Delaying action increases the organization’s risk of cyber attacks, potentially resulting in severe consequences such as data breaches, leading to reputational damage and financial loss. * What this policy may lack is sufficient emphasis on continuous monitoring and subsequent incident response. While preventive measures are essential and well thought out in this policy, early detection and sufficient response to security incidents are just as crucial for an effective defense-in-depth approach. |
| **11** | 1. Enhance monitoring and detection through implementing monitoring tools, detection systems, and security information and event management (SIEM) solutions. 2. Establish an incident response team with procedures that conduct regular incident response tests to ensure readiness. |
| **12** | 1. Invest in continuous education and training. Educate staff about cybersecurity best practices, threats, and their role. 2. Remain agile and adaptable to emerging trends in security measures. This will keep the standards evolving as threats evolve. |
| **13** | [Insert text.] |
| **14** | [Insert text.] |