# 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 Joshua Donnelly, and this is my presentation on the Green Pace Security Policy. In this presentation, I will outline the key security principles, policies, and coding standards that help prevent potential security vulnerabilities in our software development practices. |
| **2** | The Green Pace Security Policy is a crucial framework for ensuring consistent implementation of secure development practices across our growing team. The need for a defense-in-depth strategy arises from the increasing complexity of modern systems. By applying security principles at every stage, we can safeguard our software from multiple types of threats. |
| **3** | Here is the Threats Matrix, which outlines the coding vulnerabilities that could affect our systems. High-priority threats, such as SQL injection and buffer overflows, are critical to address. Automation tools like SonarQube and Fortify will help detect these issues early in the development process. |
| **4** | Next, we have the ten core security principles that guide our development practices. These include principles like validating input data, heeding compiler warnings, and designing systems with security policies in mind. Each of these principles helps ensure that vulnerabilities are minimized throughout the development lifecycle. |
| **5** | Our coding standards are essential for maintaining secure code. Here, I’ve listed the top 10 coding standards that prioritize security, such as preventing SQL injection, managing memory correctly, and ensuring secure cryptographic operations. These standards are ranked based on their impact on security and the likelihood of exploitation. |
| **6** | Encryption plays a vital role in securing our data. We employ encryption at rest to protect data stored in databases, encryption in flight to secure data transmitted across networks, and encryption in use to protect data that is actively being processed. By using strong encryption protocols like AES and TLS, we ensure that sensitive data remains secure at all stages. |
| **7** | The Triple-A framework refers to Authentication, Authorization, and Accounting. Authentication ensures that users are who they claim to be, authorization restricts access to resources based on user roles, and accounting tracks user activities for auditing purposes. Implementing these policies strengthens security by controlling access and monitoring usage. |
| **8** | Unit testing is a key component in identifying vulnerabilities early. We perform tests for potential security issues like SQL injection, buffer overflows, and memory leaks. Using frameworks like GoogleTest, we can automate these tests and ensure that any detected vulnerabilities are resolved promptly. |
| **9** | In our DevSecOps pipeline, automation plays a critical role in security. By integrating static analysis tools during the development phase, we can automatically detect vulnerabilities in the code. Additionally, continuous security checks are performed during the verification stage, while software signing ensures the integrity of releases during the deployment process. |
| **10** | Addressing vulnerabilities early brings significant benefits, such as reducing the risk of data breaches and maintaining system integrity. However, delaying action can lead to severe consequences, including financial losses and reputational damage. It's critical that we prioritize fixing high-risk vulnerabilities to avoid exploitation. |
| **11** | To close existing gaps in our security policy, we should adopt additional security standards, such as real-time threat detection and stricter access control policies. Additionally, increasing security training for developers will help ensure that best practices are followed consistently across the team. |
| **12** | In conclusion, the Green Pace Security Policy is designed to protect our systems and data from evolving threats. By implementing these security principles and practices, we can maintain a secure development environment and ensure the long-term success of our projects. |
| **13** | OWASP Foundation. (2024). *Top 10 security risks*. https://owasp.org/www-project-top-ten/  SonarSource. (2024). *SonarQube documentation*. https://docs.sonarqube.org/latest/  National Institute of Standards and Technology. (2023). *NIST special publication 800-57: Recommendation for key management* (Rev. 5). <https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final> |