# CS 405 Project Two Script

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | "Hello everyone, welcome to the presentation on the Green Pace Secure Development Policy. Today, we will discuss the comprehensive security policies, standards, and practices that we have put in place to ensure robust security in our software development lifecycle. My name is Michael, and I am excited to walk you through these crucial elements." |
| **2** | "In this section, we will provide an overview of our security policy. The Green Pace Secure Development Policy is crucial for ensuring that security best practices are consistently applied across all development activities. It defines core security principles, coding standards, and data encryption practices. This policy helps protect our software from vulnerabilities and supports a defense-in-depth strategy, applying layers of security controls throughout the development process. This policy applies to all Green Pace staff involved in creating, deploying, or supporting custom software." |
| **3** | "Next, we have the Threats Matrix, which identifies and categorizes the potential security risks in our development process. This matrix helps us prioritize our efforts and ensure that we address the most critical threats first. By understanding the likelihood and impact of each threat, we can allocate resources effectively and implement the necessary controls to mitigate these risks." |
| **4** | "Here are the ten core security principles that form the foundation of our policy. These principles guide our approach to secure software development and ensure that we maintain a strong security posture. Each principle aligns with specific coding standards, helping us implement these best practices consistently across our development efforts. The principles include secure design, least privilege, defense in depth, secure defaults, fail securely, separation of duties, keep it simple, avoid security by obscurity, security as a process, and psychological acceptability." |
| **5** | "Let's delve into our C/C++ coding standards. These standards are essential for maintaining code quality and security. They help prevent common vulnerabilities and ensure that our codebase remains robust and maintainable. We have prioritized these standards based on their impact on security and code integrity. The detailed explanation of each standard follows." |
| **6** | "Here are the first five detailed coding standards. Standard one ensures the correct and appropriate usage of data types. Standard two focuses on maintaining consistent and valid values for data variables. Standard three emphasizes the use of proper string handling techniques to prevent buffer overflows. Standard four aims to prevent SQL injection vulnerabilities. Standard five ensures effective memory management to avoid memory leaks and other related issues." |
| **7** | "Continuing with our coding standards, standard six ensures proper exception handling. Standard seven advocates for the use of secure communication protocols. Standard eight focuses on validating and sanitizing input data to prevent injection attacks. Standard nine emphasizes simplifying code implementation to reduce complexity and potential errors. Finally, standard ten promotes the use of effective quality assurance techniques to maintain high code quality." |
| **8** | "Our encryption strategy covers three critical areas: encryption at rest, encryption in flight, and encryption in use. Encryption at rest protects data stored on disk from unauthorized access. Encryption in flight ensures that data transmitted over networks is secure and protected from eavesdropping. Encryption in use protects data while it is being processed. These policies ensure that our sensitive data remains secure throughout its lifecycle." |
| **9** | "The Triple-A framework stands for Authentication, Authorization, and Accounting. Authentication verifies the identity of users and systems to ensure that only authorized entities can access our resources. Authorization grants access rights based on verified identity, ensuring that users can only access the resources they are permitted to. Accounting involves recording and auditing access and actions, providing a trail of activities for security monitoring and compliance." |
| **10** | "Unit testing is a critical part of our security strategy. We use unit tests to identify and mitigate coding vulnerabilities early in the development process. Each test addresses a specific vulnerability, and we apply the unit testing frameworks for C++ in Visual Studio to ensure thorough coverage. These tests include a mix of positive and negative scenarios to validate the robustness of our code." |
| **11** | "In this section, we summarize our automation strategy. The DevSecOps pipeline integrates security tools throughout the development process. These tools reside in various stages, including coding, building, testing, and deploying. By automating security checks, we can detect and address vulnerabilities early, ensuring that our code remains secure throughout the development lifecycle." |
| **12** | "Here, we discuss the risks and benefits of our security strategy. Addressing vulnerabilities proactively helps prevent potential security breaches and reduces the overall risk to our organization. However, delaying action can lead to increased risk and potential exploitation by attackers. By acting now, we can mitigate these risks and strengthen our security posture." |
| **13** | "In conclusion, we have conducted a gap analysis of our existing security policy and identified areas for improvement. By adopting new standards and practices, we can prevent future problems and enhance our security framework. These recommendations are based on industry best practices and the latest security research." |