# CS 405 Project Two Script

[Video Link](https://www.youtube.com/watch?v=qqrXQEe-_pI)

Kyle Blan 4/28/24

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Welcome everyone, and thank you for tuning into this presenation that is all about best practices in secure coding, and adhering to coding standards. |
| **2** | In this presentation, we'll explore the importance of secure coding standards and their role in establishing a robust foundation for secure software development. This is further illustrated by the defense in depth diagram. By implementing this policy, we establish a framework for secure coding practices that permeates throughout our development processes. |
| **3** | Now, let's delve into our threat matrix, a vital tool for identifying and prioritizing potential security threats. Our threat matrix categorizes security threats into four main categories: Likely, Priority, Low Priority, and Unlikely. Each threat is evaluated based on its likelihood and priority level, allowing us to focus our security efforts on the most significant risks. |
| **4** | In this section, we'll explore the ten principles of secure coding and their alignment with our adopted standards. As you can see, our policy includes a heavy emphasis on software methodology and quality assurance practice. |
| **5** | Here we will explore the ten adopted standards that form the backbone of our secure coding practices. These standards provide guidelines and best practices for writing secure, resilient code. |
| **6** | Now, let's discuss our encryption policies, which play a crucial role in protecting sensitive data from unauthorized access and ensuring its confidentiality and integrity. It is important to note that by implementing these encryption policies, we can strengthen our data protection measures, mitigate the risk of unauthorized access, and ensure compliance with regulatory requirements. |
| **7** | Now, let's delve into the core of our cybersecurity strategy—the Triple A policies of Authentication, Authorization, and Accounting. These policies form the backbone of our defense mechanisms, ensuring the integrity, confidentiality, and availability of our organization's resources. In essence, the Triple A policies serve as the guardians of our digital fortress, fortifying our defenses, and preserving the trust and confidence of our stakeholders. |
| **8** | Here we have our Unit testing results, as you can see the testing was entirely successful. |
| **9** | We have our first test case that was simply labeled “Always Fail,” and is designed, of course, to intentionally fail. |
| **10** | More information about our unit testing provided here, verifying that data can be added to a vector. |
| **11** | Unit testing is continued here, further information about a specific case, where we add 5 values to a vector. |
| **12** | Finally, we have a test case here that shows we can throw proper exceptions when a value is out of range. |
| **13** | DevSecOps aims to integrate security throughout the software development lifecycle, ensuring secure code delivery, and in this case integrating security practices throughout the software development lifecycle (SDLC). |
| **14** | Next we have a tool summary showing how different toolbox items help to achieve a sager SDLC. Our DevSecOps pipeline is made up of component parts, each of equal importance. |
| **15** | In summary, we've identified a critical problem: the lack of adherence to coding standards. This issue can manifest in various ways, including inconsistent code quality, increased vulnerability to security threats, and difficulties in collaboration and maintenance. Fortunately, we have a solution at hand: implementing coding standards. By adopting guidelines and best practices for writing secure, maintainable, and efficient code, we can address these challenges head-on. |
| **16** | As for recommendations, our policy covers only a narrow scope of security sensitive topics. To ensure a secure pipeline, we must implement a full complement of coding standards to govern all aspects of the SDLC. |
| **17** | First and foremost, Code Review and Testing Standards play a vital role. These standards ensure that software undergoes thorough review and testing for security vulnerabilities, logic errors, and other issues before deployment. By doing so, the likelihood of bugs and vulnerabilities in production is significantly reduced. Next, Access Control Standards are essential. These standards enforce the principle of least privilege, ensuring that users are granted access only to the resources and functionalities necessary for their role or job function. This minimizes the risk of unauthorized access and data exposure, bolstering overall security posture. Finally, Logging and Monitoring Standards are indispensable. By implementing these standards, organizations can track and analyze user activities, system events, and security incidents in real-time. This enables early detection and response to potential threats and breaches, enhancing the organization's ability to safeguard its assets and infrastructure. |
| **18** | If you have any questions about the presentation, please feel free to drop a comment down below. |