# Shanine Efferson

06/29/2025

Project Two: Security Policy Presentation

https://youtu.be/sAzfI5XjkLc

# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Welcome, my name is Shanine, and I will be presenting the brand-new Green Pace security policy guide. This presentation will provide an in depth look into the guide to ensure our development team remains in sync with the principles and best practices. |
| **2** | This security policy was created to define the secure coding principles, standards and encryption that Green Pace developers must follow and incorporate into future tasks. The security policy will focus on protection through DID (defense in Depth), which is a layered security measure. |
| **3** | For the threat matrix the security risks are broken up into there threat levels.  SQL injection is a priority risk because it can cause critical breaches.  Buffer overflow is common and should be monitored closely  Code formatting is a low risk but could cause some errors  And unlikely risk would be supply chain attacks. |
| **4** | These are the 10 coding security principles, each principle helps use address specific security risks. |
| **5** | I listed the 10 coding standards in priority order based on severity, how likely the threat is to occur, and cost to fix. This ranking will allow the team to focus on the highest priority and still enforces the guide. |
| **6** | Encryption at rest protects data stored on physical devices like, hard drives, databases and other system backups. Data cannot be stolen and is protected from security breaches  Encryption in flight: Protects data as it moves from one system to another. This protects data such as user logins and communication from being intercepted.  Encryption in use: Protects data while being used in a program, this is used because data may become vulnerable when loaded into memory. |
| **7** | Here is the triple A framework  Authentication is the systems way of verifying a users identity  Authorization decides what the user can access and Accounting,  Keeps track of everything the user does by logging actions. |
| **8** | This test checks if a newly created collection is empty. To start, a new collection was created, and a test was ran to check if it was empty, this was then verified that the collection size was 0. This is a positive test since the collection did what was expected. |
| **9** | This test checks what happens when one value is added to an empty collection. The result passes, making the new collection have a value. |
| **10** | This test checks if the vector’s max size is greater or equal to the current collection size, the test passed and does not create an error. |
| **11** | This test is a negative test, meaning and error is purposely thrown when ran. Because an out of range exception was successfully thrown, the test passed. |
| **12** | This diagram shows how DevOpsSec will work between the pre production and production of a system. This provides security at every stage of the programs life cycle. |
| **13** | Green pace is changing its DevOps approach into a DevSecOps. This means that security will be apart of each step of the process. External tools like. Parasoft and Klockwork will be used in the pre production phase as shown in the diagram. This allows problems to be caught and fixed early on. During the production phase, runtime tools like CodeSonar, will be used. |
| **14** | The current security system is missing secure coding and real time monitoring  To fix this we can strengthen automation, and expand testing  If we wait to apply these changes we can face breaches and issues with company compliances |

|  |  |
| --- | --- |
| **15** | There are some gaps within the current policy. There are no runtime threat tools, there are a limited amount of testing and inconsistencies with the coding practices. I recommend adding runtime tools, provided training to the development team and making frequent coding tests and reviews. |
| **16** | In conclusion, to prevent future issues we should follow the SEI CERT C++ standard, this will allows developers to stay consistent and adopt a strong security mindset. |
| **17** |  |
|  |  |
|  |  |