# CS 405 Project Two Script

Author: Anthony Lee

December 11, 2022

Project Two: Security Policy Presentation

Video link: <https://youtu.be/3vVEuDJBA7I>

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Welcome to my security policy presentation. My name is Anthony Lee, and I will be showcasing for everyone at green pace what our security policy is |
| **2** | One major topic to cover right from the start is defense in depth, or the philosophy that information security mechanisms are carefully and thoughtfully layered. This will create strong protection against risks and possible vulnerabilities. It is always better to avoid trouble than get hurt by an attack and require a band aid to fix it. |
| **3** | As outlined in the security policy document provided for you, here are 10 coding standard threats and their associated risks. Without going into too much detail, please note that there are a lot of high severity threats that need to be addressed |
| **4** | To combat these threats, here are 10 principles of secure coding outlined in the security policy document and the associated threats for them. Please note while there are some principles not listed with threats, they are still important to follow as there are more threats than just the 10 listed |
| **5** | This slide is the coding standard and their associated threat level broken down into a table for easier digestion, also including their cost for remediation, priority, and level of importance |
| **6** | There are three encryption policies that need to be followed for this security policy to function: encryption in rest, at flight, and in use. Encryption in rest is protecting stored data that is not in use, such as the information stored in your phones and laptops. Encryption at flight is protecting data that is being transferred, either physically or electronically, such as sending emails. Finally, encryption in use protects data that is being currently used, such as reading and modifying documents. |
| **7** | Triple-A policies is another aspect of this security policy. The As stand for authentication, ensuring a person is who they claim to be; authorization, making sure what a user can and cannot access within a system; and accounting, or the process of logging and account for all digital actions to maintain the well-being of the system and its data. For more information, you may look at tech target’s article on Triple-A policies |
| **8** | Unit testing, or the process of verifying and validating components of a program, is important to do during development to ensure the application will act and respond as intended. We use Google Test, or gtest, for this. The following slides are a few examples of unit testing showing a passing result |
| **9-13** | 1. This test is checking if we can add anything to an empty vector. It is important to know where or not we can add data before moving forward 2. This test checks if our collection can be increased. To check, we populate a vector, create a variable to hold the current size, and resize it. Then we see if our new size is bigger than the previous one. 3. This test checks if we can actually clear all the entries to empty our collection. It is important to clear data for example removing a person from our records so they do not have their credentials saved after leaving the company 4. This is a test we are trying to fail by trying to read an entry that does not exist. We throw an exception because entry number 20 does not exist in this vector of 5 entries. This is important to prevent injection attacks and other issues with protecting your data 5. Here is another test we are trying to fail by trying to remove an element and looking for an element outside of the range of our vector. Once again this helps prevent errors caused by looking out of range |
| **14** | Automation is an important tool you can use at Green Pace to create secure code with low overhead costs. This cycle is called the DevSecOps |
| **15** | According to redhat.com, DevSecOps, stands for development, security, and operations. It’s an approach to culture, automation, and platform design that integrates security as a shared responsibility through the entire IT lifecycle. Starting in pre-production, tools like CPPCheck are used for static code analysis to find problems in the code. A compiler that has tools for automating this is Clang. Parasoft, which an automated testing suite, paired with these tools makes automating DevSecOps easy to implement into your SDLC. |
| **16** | There are many risks involved by waiting to implement security. Remember, it is always better to be prepared than not. Waiting can cause harm to you at Green Pace by harm to data, harm to your reputation and customer trust, financial cost associated to the damage, and the future damages to your company to multiple attacks of the same nature. The benefit of early implementation is it makes it easier to mitigate the damage, prevents possible attacks, makes it easier to be consistent with security changes, and reduce testing overhead. [read example case] |
| **17** | This policy is the start and foundation of security at Green Pace  However, every policy needs to be subject to regular review and updates as threats are constantly evolving and improving  Waiting for an attack before acting leads to costing more than the upfront cost of being prepared beforehand  Annual checks from an outside, third-party source, such as a white hat security firm, will give the security a real-world test to find possible vulnerabilities  To ensure security is consistently at the forefront of progress, do not leave security until the end of development. Instead, early application of these policies will promote better security |
| **18** | Adopting best practices and coding standards, defense in depth, constant consideration to potential attacks and their motives, and keep a “zero-trust” policy towards security and prevention throughout development will help to transform DevOps into DevSecOps at Green Pace. |