# CS 405 Project Two Script

Steven Edmund

Project Two: Security Policy Presentation

December 10, 2022

YouTube: <https://youtu.be/J269QO2_oIc>

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello, my name is Steve Edmund, and I will be presenting the new Green Peace security policies.  So, without further ado, here is a quick rundown! |
| **2** | An idea of defense in depth is presented. Defense in depth, as the name implies, is an approach to cybersecurity that employs multiple layers. There are multiple layers of protection, each with its own set of requirements. When one layer is breached, the one below it provides protection from further assaults, much like redundancy. It offers metrics for a wide range of processes and even hardware. Therefore, it's a very effective policy. |
| **3** | Threat Matrix  The threat Matrix is a means to see the potential dangers of a certain coding standard or style. Risk is measured in terms of its occurrence, impact, remediation expenditures, and complexity. The degree to which one coding method mitigates risk will determine its relative importance. |
| **4** | Here are the top 10 principles of coding. Here, we make reference to Kirsh's Top 10 Secure C++ Coding Practices article. |
| **5** | I have come up with certain coding standards, and they include common errors made by programmers in the field who aren't always conscious of the repercussions of their actions. |
| **6** | We are thinking of adopting a "zero trust" encryption policy, which means that nobody can be relied on to keep messages private. In these cases, any of the three possible encryption methods may be used.  The first one is Encryption at flight. In other words, it is the act of encrypting data before sending it across a network. Since the sending entity (the organization) has little to no say over the receiving entity (the network), transmission is the oldest and most reliable element of security. This means that the vast majority, if not all, of communications, are encrypted.  The second one is Encryption at rest. The disk-based data encryption is the main emphasis. When employees have access to a secure storage facility, data leakage is reduced. One way to do this is to guard the encryption key carefully.  The third one is Encryption in use. This section works to make some elements of the storage inaccessible to the user throughout their session. Any sensitive information, such as a secret key, has to be protected from prying eyes. |
| **7** | Triple-A Policies  Authentication, Authorization, and Accounting are sometimes abbreviated as Triple-A.  The authentication process determines who is allowed to access the system. In order to do that, we provide each user with a unique identifier and password.  The authorization phase determines which services the authenticated user is permitted to access. To provide one example, administrators will have more power than regular members.  Accounting is the process of keeping track of users' actions for the sake of later review, as well as throttling and intrusion detection. |
| **8** | Here we see the results of a float loop test. One should avoid counting using float. If the accuracy error is too great, we may never achieve the maximum or lowest value, and the process will continue indefinitely. |
| **9** | In this slide, we see an example of a string that hasn't been ended, which is a typical error made by developers that may lead to serious issues. |
| **10** | You can see the unsigned integer wrapping test in action on this slide. When an unsigned integer is smaller than zero, and we subtract from it, the result is positive rather than negative because the integer wraps around to its maximum value. |
| **11** | In this slide, we see the hanging pointer scenario. When deleting or freeing a variable, it is recommended to set the variable's pointer to NULL. |
| **12** | The DevSecOps process is shown in this figure. The graphic makes it easy to understand how DevSecOps incorporates security checks across the whole software development process. |
| **13** | The DevSecOps pipeline is a method for incorporating security best practices into the design phase of software creation. This ensures that the program is solid from the beginning.  Compilers, debuggers, and testing frameworks are some examples of additional tools that may be used to help reduce security risks. |
| **14** | Risk and Benefits  We can only reduce security risks as far as the benefit-cost ratio will allow. There is no use in trying to eliminate risk if doing so would cost more than the danger itself.  However, it is imperative that we underline the need to lessen the threat to security. My view is that the combination of DevSecOps and defense in-depth is quite strong in terms of reducing risk. |
| **15** | Although rules may be formulated, it may be difficult to convey them to programmers. That's why it's helpful to maintain policies consistent and easy to understand. Rather than having specialists check in on the system every once in a while, we should make the security principles consistent. |
| **16** | So, to sum up, it is indeed clear that combining development and security principles helps produce more resilient software. In addition, not putting any faith in any of the involved parties but rather adopting a zero-trust communication strategy and striving to fix any problems that may arise may make the system very secure. |
| **17** | These are the references that I used for this presentation. Thank you! |