# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | This is my presentation about Green Pace Security Policy. |
| **2** | As we can see this illustration shows Defense in depth, it shows the different layers in the defense system, it shows how a program should be protected like the IRL defenses so it can protect the system from any attack. this illustration shows the different types of components of physical security and as well network. |
| **3** | This next slide shows us how the threats are by how vulnerable they are. The table show the following vulnerabilities likely, priority, low priority and unlikely. |
| **4** | [this slide show s the 10 principles whenever you are considering if your security is good or not.  By validating the input data, we ensure only the right information enters the system. This means that any bad and malicious input data can be prevented. In addition, this means that we can verify the accuracy and the integrity of the data that’s being input in the system. So, then we can avoid any data corruption, unauthorized access and any system crash that can occur. In addition, by implementing the right validation for the user input like data length, validation range etc.  The Heed warnings allow the developer to know that there is an error or issue within the code. This means as well that there might be a buffer overflow within the project. There are many ways to reduce risks, overflow etc. and some of them are using the right IDE and enabling the basic runtime checks (/RTCs) within the C/C++ options will allow compiler generated runtime checks for buffer overflows.  Ensuring that the right security is implemented in the software is necessary using the best practices for security policies, such as separating a system into sub systems with different authorization or privilege levels. This ensures that the right people have the right security clearances this will prevent any unauthorized people to access the system.  Having a simple design allows you to reduce any risks in the codes and whenever the users are using the system, this as well means that it helps with the minimalization in the complexity of security required. This means proper spacing, proper variable names and useful comments always make sure to follow the KISS rule. Make it short and simple.  Using the default deny this allows that when the access and access is permitted through the conditions of the protection scheme deems that permission is authorized. Either by firewall policy as well blocking all inbound and outbound traffic that has not been permitted expressly by the policy.  This will help to reduce a big risk, and this is because allowing the user to have the necessary levels to complete their task without giving them too much power, this means that. MYSQL is able to employ the principle of least privilege when several user accounts are each designated toward their own unique DB task and privileges. This reduces any risk of unauthorized users to have unnecessary privileges.  By sanitizing the data that is being sent to other systems this can help avoid any injection attacks or threats to the systems. This allows to check for potential issues prior reaching the system by using the right data sanitation that includes a parameterizing an SQL query to avoid SQL injection. Two main approach that can be taken into consideration would be black-listing and white-listing these two can help with the vulnerability of the system for example by using a deny by default approach would be to white-listing, this is a list that only valid inputs are accepted, allowing fail-safe behavior on unexpected inputs.  Ensuring using multiple layers would help with avoiding any risks. This means that if one layer was breached it has plenty of layers left that would secure the system. These layers of protection could be firewalls, intrusion detection, antivirus/antimalware, VPN or even Virtual Machine. We could add as well maintaining the servers continuously, using harden code and crypto library as well can be of big help to prevent any SQL injection.  There are many ways to ensure the code is reliable, secure and that can avoid unnecessary bugs and is by doing the necessary quality assurances techniques like penetration testing, vulnerability scanning, code reviews, security audits, and vigorous testing etc. can help with quality assurance techniques.  Ensuring you are applying the right methos to the code can help a lot as well as a secure coding standard is a must in any project this helps secure any standard that Is being adhered to SEI CERT C++ Coding Standard. This not only applies to C++ or SQL but as well any other coding languages |
| **5** | The coding standard table show us ways to protect our system as well it shows the priority, the level and the severity. |
| **6** | Slide number 6 shows us the different types of encryption policies and their description. It explains why they are important and how and where to use them. |
| **7** | The triple A policies talk about authentication, verification and accounting this will help with ensuring your data is not being stolen or the account and private information that someone doesn’t need access so it can be secured. |
| **8** | The unit testing slides starting from slide 8-11 talk about the different unit testing. Google tests help us test for independent and repeatable tests.  Slide 8 tests for MaxSizeGreaterThanSize  Slide 9 tests for ResizeIncreasesCollectionSize  Slide 10 tests for ResizeDecreasesCollectionSize  Slide 11 tests for CanAddFiveValuesToCollection |
| **12** | The automation summary explains how we can maintain and secure our applications as well how automation can be created upon Build, by automating manual processes like static code checking into a CI/CD pipeline.  By verifying and testing in SecOps this helps with implementing good security testing.  Monitor and detect helps with security testing into specific builds of code, as well we could add that embedded dynamic application security helps with looking for vulnerabilities. |
| **13** | Slide 13 offers different types of tools to keep your software secure. |
| **14** | Slide 14 explains the risks and benefits. Don’t leave for later what you can do right now it can be costly and you can lose revenue, customers and more. Starting early from the design phase is a good thing to do because this ensures that the security goes through the right testing [phases it needs to go through. Lower the risks of any issues that can arise. |
| **15** | Slide number 15 talks about the best recommendations to ensure we provide the best security on how to ensure to lower security risks, malware and APIs. |
| **16** | In conclusion by following these securities policies any breaches or issues that can come up it can help mitigate the damaged before it can grow bigger. The more the issue is left for later the worst and the bigger the damaged can be as well the bigger the cost is going to be. |
| **17** | There are the resources that were used in this project. |