# CS 405 Project Two Script Template

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

YouTube Video Link: <https://youtu.be/wpyEVG9x5cI>

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello. My name is Aamir Rahman. I am a developer and a student at SNHU. This is my security policy presentation for Green Pace for course CS405 Secure Coding |
| **2** | There are several layers that exists for defense in depth. Each layer has a main purpose to protect sensitive information. The idea is that if one layer fails, the next layers will step in to protect the sensitive information. The problem is the number of layers. Having too many layers can also create a mess. Having too many layers can also be expensive and maintaining theses layers can also be an issue. The number of layers will depend on the organization that’s implementing these security measures. It’s very crucial for all orgs to have a very good defensive system in place. As fast as technology is growing, there are also hackers with various tools and strategies to create havoc. The diagram on the left shows the various defensive layers and the different ways to protect a system. Organizations can use many of these mechanisms to protect company data. |
| **3** | The table on this slide is a threat matrix. This threat matrix shows how likely it is for threats to occur and their priorities. Likely type of threats has a high chance of occurring and frequently. Unlikely threat type is the opposite of likely, so unlikely threats are threats that won’t appear often, but can also occur. Priority identifies threats that should be at the top of the list and needs to be fixed RIGHT AWAY. Low priority on the other hand are issues that are not detrimental, but should still be fixed when possible |
| **4** | This slide shows the 10 principles of security that Green Pace must follow.   1. Validate Input Data 2. Heed Compiler Warnings 3. Architect and Design for Security Policies 4. Keep It Simple 5. Default Deny 6. Adhere to the Principle of Least Privilege 7. Sanitize Data Sent to Other Systems 8. Practice Defense-in-Depth 9. Use Effective Quality Assurance Techniques 10. Adopt a Secure Coding Standard |
| **5** | I listed the 10 coding standards on this slide, and it’s ranked from the highest to the lowest severity. I also provided a table that shows the severity, likelihood, cost to remedy, priority and levels. |
| **6** | This slide shows the encryption policies.  Encryption in Rest:  This is where data is being encrypted while the data is being stored. Where unauthorized users cannot decrypt it and thus preventing the data from falling into the wrong hands. It’s converting the data into another form of data through an algorithm that can only be understood by a person who has the key in order to decode it. This policy applies because it helps protect the data in the system. Helping defend from hackers getting their hands on sensitive information of customers and/or the company.  Encryption at Flight:  This is where data is being encrypted while it is transmitting over the network. This helps prevent getting data stolen over the network. Each side, sender and receiver (if authorized) will have an encryption key that is unique so that once they have the data, it can be de-encrypted. This policy applies because it helps from preventing transmitting data stolen.  Encryption in Use:  Data that is either at rest, in use, or in transit will never be left unsecured. It does not matter where it is being stored but the sensitive data will always be protected. This policy applies in order to keep sensitive data, regardless on what it is doing (at rest, transmitting, or being used) is always secured to assure the customers, partners, etc. that their information is safe. |
| **7** | This slide shows the tiple-a-policies in place.  Authentication:  Authentication is necessary to provide the business a way to allow certain users to access the system. This can be done using a user ID, password, and possibly a biometric signature. This policy applies because it prevents unauthorized users out of the system and accessing data.  Authorization:  Authorization is needed to grant access/rights for the authenticated users. This allows for limiting users from accessing information that they do not need, and also granting users to accessing information they do need. Data that is being accessed by too many users when only a few should be accessing it can make the data less secured.  Accounting:  Accounting is basically a checks and balances for a business. Making sure the system is being monitored and being aware of the data that is being used. It also helps with authentication and authorization to see who has what capabilities and if they should keep said capabilities. It can help in preventing data getting into the wrong hands. |
| **8** | This unit testing verifies that resizing decreases the collection. I initially added 15 entries, and then initialized and declared previous value, resized the container to hold an element, and finally tested to find out it’s true. |
| **9** | This unit testing verifies that resizing increases the collection. I initially added 1 entry, and then initialized, resized the container to 15, and finally tested to find out it’s true. |
| **10** | I continued to unit test with different attributes for the next two sides. This slide I tested to verify resizing decreases the collection to zero. |
| **11** | This slide I tested to verify reserve increases the capacity but not the size of the collection. |
| **12** | Finally, this slide shows the overall results of all of the specific unit tests. Last few slides only showed four, but there were other scenarios of unit testing that was executed, and this shows the results. |
| **13** | This slide shows the automation summary. Automation is used to enforce the standards that are defined in this policy. |
| **14** | DevSecOps is where security is integrated into your software development life cycle. In order to build, test, and deploy secure software faster, there needs to be some security practices put in place. This helps with detecting security vulnerabilities in the early stages and a faster recovery speed if a security incident were to occur (Foster, 2021).  An external tool that can be used is cppchecker. This tool can go line by line of the warnings/messages that need to be addressed. In my experience, cppchecker caught more issues than Visual Studios did. In a coding project, Visual Studios marked 2 lines with errors, while cppchecker marker over 15 lines. This can help fix and secure code. |
| **15** | This slide will review the risks and benefits when it comes to adding security features to your project. There are two different timelines, act now and act later.   * Act Now   + Mitigates any current threat risks   + May save money in the long run by preventing any imminent attacks   + Secure people’s information quickly / Keep the trust of our clients   + May not have all the resources to act now   + High upfront cost * Act Later   + Time to retrieve resources and money   + Attacks may happen in the meantime   + Lose trust of clients if a data breach occurs   + Could cause a bigger, more expensive, issue in the long run |
| **16** | Here are some recommendations for increased security.   * + - Follow security standards     - Encryption Policies     - Revise and Edit Policy Regularly     - Reference All CERT Coding Standards     - Start Implementing Encryption and Triple-A policies as soon as possible     - Teamwork     - Security Training     - Start Internal Defense-in-Depth Layers |
| **17** | Technology is growing at a rapid pace. More software, more coding, more tools, and more technology. Technology definitely makes our lives easier. But with increased technology, there are increased risks of security concerns and vulnerabilities. As technology increases, hackers are becoming smarted to find ways to hack into these systems and breach. To avoid any breach in a system it’s important to follow the security standards in place such as the ten principles, triple-a-policies, and encryption policies. There’s also Defense in Depth in place (DiD), thorough testing, and security procedures in place to improve security when other methods fail. Security policies will have to be enforced by the organization and to every member of the organization to make sure everyone is following it as expected. |
| **18** |  |