# CS 405 Project Two Script Template

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

Josh Ryther

Project Two: Security Policy Presentation

Youtube Link: https://youtu.be/mrkIJOVXDlU

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello everyone. My name is Josh, and I am a software security analyst for Green Pace. I have been tasked with developing a new security policy for the company and will present the details necessary to implement it into future projects. |
| **2** | The security policy was developed out of a necessity for core security practices to be implemented into our day-to-day. Attacks come from all angles which will find and exploit our weakest link. We must nurture a defense-in-depth mentality to provide security at all layers so that if one vulnerability is found hackers will not have unrestricted access. |
| **3** | The threat matrix pictured pulls four rules from the policy that represent standards that should be implemented by priority and likelihood of occurring. Priority targets have a high risk of vulnerabilities and if they are likely to happen then the damages could be significant. Automation software and techniques should be utilized to detect and resolve these cases when possible. |
| **4** | The 10 principles listed are the practices that will go into effect. Each one is important to maintaining top notch security. For those that have a relevant standard I have listed it next to the principle as an example of how it would be implemented. |
| **5** | The 10 standards have been displayed in a table and are ranked by priority. The priority is determined by severity, likelihood, and remediation cost. Remediation cost has been left out of the table for readability. These factors influence the potential costs associated with mitigation and recovery from an attack. |
| **6** | The three forms of data encryption are important to securing sensitive information used by our applications. All data at rest must incorporate some level of encryption. Even non-sensitive data must have a base level of encryption to obfuscate the stored information. When using public Wi-Fi, a VPN must be utilized to maintain encryption at flight. Data used by employees on a public network is open to interception and should be secured. Lastly, credit card numbers need to be encrypted at all points of the application which includes while in use. This data is highly sensitive and should allow no points of decrypted access by attackers. |
| **7** | Triple-A policies ensure that users are confirmed before access, limited in scope to their respective roles, and tracked to maintain accountability. Authentication will be maintained by requiring Google 2FA for access to the system. This will create a higher level of authentication beyond a simple password. Once logged in each job level will be assigned an authorization role that allows the least amount of access required to perform daily duties. Lastly, all changes in the system will be logged and stored for 7 years. This level of reporting will provide better data to find and prevent unwanted actions performed by users and programs. |
| **8** | The next four slides show examples of unit tests that when utilized correctly can enhance the security of an application. The first test confirms that the function clear empties the vector. The next test checks that the reserve function increases the capacity of a vector but not the size of the collection. The third slide checks that the resize function decreases the collection to zero. Lastly, the fourth test makes sure that the out-of-range exception is thrown when an index is out of bounds. Utilizing all four of these tests on a vector can ensure that the functions are working correctly and decreasing the chances of undefined behavior. Unit testing should strive for 80% code coverage in order to maintain maximum stability. |
| **9** | Automation flow diagram |
| **10** | Security is ingrained at all stages of the software development lifecycle in DevSecOps. Static testing can be used to determine design vulnerabilities before coding begins. During QA there are security focused frameworks that can test for common vulnerabilities along with faults in the software. Lastly, logging tools can be utilized to provide accounting to monitor the behavior of the program in production. That data can then be used to influence changes to security in the next sprint cycle of features. |
| **11** | With any decision in leadership there are risks and benefits to the choices that are made. Focusing only on security will increase the stability of the application but will force the project to miss out on more features. New features will add an increased risk of vulnerabilities but will be more enticing to users. Leaders will need to balance the amount of security implemented versus adding capability to the product. |
| **12** | The new security policy being implemented is very secure but must continued to be analyzed for potential gaps and improvements. Current issues that have been identified and will be remediated in future versions are increased standards for common cases of buffer overflow, exception handling, and input validation. |
| **13** | Three examples of future standards to be implemented are string character limits, custom class exception handling, and input validation for integers, string, etc. Buffer overflow can occur in several data types and was only touched on once. Exception handling of third-party libraries is important but must also extend to in-house classes and programs as well. Custom exceptions must be implemented and handled to ensure unexpected behavior is addressed. Lastly, input validation must be utilized and standardized for each data type. Allowing user input can cause many undefined issues and become a security risk. |
| **14** | Thank you to everyone that has attended the presentation. I hope you better understand the security policy going in effect and if there are any further questions, please feel free to reach out. |