# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello Green Pace, my name is Nathanael Burke. I will be providing a security policy presentation to you today. |
| **2** | Within this security policy we will be discussing the various elements that make up the security for Green Pace as well as educating the team on why it is important and why it is needed. I’m going to start first with explaining how this policy correlates to Defense in Depth. Then I will walk you through some of the principles, coding standards, and encryption policies I have followed by explaining the importance of testing. I will then end the presentation with Some risks and benefits and recommendations I have for Green Pace.  The primary need for the security policy was that Green Pace currently is lacking in security. There may be security here and there, but we are here to buff the security you have and provide you with a strategy going forwards. This security policy satisfies the concept of defense in depth as it provides multiple layers of defense to your security. |
| **3** |  |
| **4** | On this slide I have listed all of the ten core security principles. Next to the principle you will see a number. This number correlates to the corresponding coding standard that uses this principle.  The first principle is to validate input data and it states to validate input data from all untrusted data sources.  The second is to heed compiler warnings and it states to compile your code using the highest warning level available on your compiler and to work to eliminate these warnings by modifying the code. Coding standard 7 utilizes this principle.  The third is to architect and design for security policies and it states to create a software architecture that is geared towards enforcing security policies. Coding standards 1, 6, 8, and 9 utilize this principle.  The fourth principle is to keep it simple, and it is just as straight forwards as it sounds. You should always strive to keep your code simple. Coding standards 3 and 8 utilize this.  The fifth principle is to default deny and it states to essentially default your system to deny access.  The sixth principle is to adhere to the principle of least privilege and it states to write your code where every process should be able to execute with the least set of privileges necessary in order to complete the job.  The seventh principle is to sanitize data sent to other systems and just like principle four it is exactly as it sounds. You should sanitize any data being sent to subsystems. Coding standard 4 utilizes this principle.  The eight principle is to practice defense in depth. As mentioned before, this involves creating multiple layers of security so that if one layer is broken another is in its place to prevent further attack. Coding standard 10 utilizes this principle.  The ninth principle is to use effective quality assurance techniques and it states to include various forms of quality assurance techniques such as testing and source code audits to identify and eliminate vulnerabilities. Coding standards 2 and 9 utilize this principle.  Lastly, the tenth principle is to adopt a secure coding standard and it states to develop a secure coding standard for your project. Coding standards 1, 5, 8, and 10 utilize this principle. |
| **5** | On this slide, I have listed the ten coding standards that I addressed in my security policy. These ten standards include data type, data value, string correctness, SQL injection, memory protection, assertions, exceptions, memory management, characters and strings, and input output. I have gone ahead and included the label and the name of the standard that I have associated with each standard type.  You can also see at the end of each of the standards I have included a level of vulnerability. The way that I determined the ranking for each of these was based on their severity, likelihood, and remediation cost. Many of these standards had varying degrees of priority and this helped to determine an overall level of vulnerability. |
| **6** | Encryption is an important part of security; it has multiple different uses and is vital to the protection of data when it is at its weakest points. Encryption at rest deals with encrypting information that is stored on the computer/server’s disk. The policy on this would be to encrypt this information at this location in order to prevent security breaches and data leaks.  Encryption in flight is about encrypting information that is being sent from one location to another. This can be from the user to the database and back or even through things like emails. The policy here would be to encrypt any information that is in transit from point a to point b.  Encryption in use is about encrypting information that is being stored. This means, and the policy reflects this, is to encrypt any information that is being stored in a database or any other data storage location. |
| **7** | Triple-a stands for the security group of authentication, authorization, and accounting. Each provide layers of defense and are important to the overall security of the system.  Authentication focuses on verifying the identity of individuals using the software as well as requiring these individuals to have authenticate any changes they make or any files they try to access.  Authorization limits users to what they are able to accomplish within the system based on their job’s necessities. This also keeps higher priority areas secure by requiring a higher form of authorization.  Accounting involves being aware of attacks on the system and taking measures to prevent further attack. Accounting also involves using routine security checks and other security audits in order to prevent possible breaches. |
| **8** |  |
| **9** | This is a diagram of the DevSecOps pipeline. I will provide a further explanation of what the DevSecOps pipeline is and how it is used in the next slide. |
| **10** | In the previous slide we saw the DevSecOps pipeline. How this operates is very similar to the DevOps pipeline, however, security is addressed and considered at every step of the way. We would start this process by assessing and planning then move into designing the code. Here we would take a look at potential security threats and what we can do now to address them. Next, we move into the building, verifying, and testing phases. These can happen in tandem in the form of unit testing and overall as system testing.  From here the product would move into production from staging and would be launched. When launching a product there a couple of locations and things to keep an eye on to make sure that everything stays operational. These include the system health and monitoring for any threats or breaches. If a breach does occur, then we would move into the response phase followed by a period of stabilization and maintenance. Once that is over, we would start the pipeline again on how to better design the system.  Some external tools that could be used include static analysis tools and other testing materials that will either automate testing or show vulnerabilities in the code. |
| **11** |  |
| **12** | For potential recommendations, I will always recommend performing routine security checks and other system analyses in order to verify the security of the system. I also recommend when looking at the system to think about what the motive behind an attack would be. Sometimes understanding what an attacker may be after may help to form a better idea of what needs to be more secure. This in turn can spark conversation on how to create more in depth security. |
| **13** | In conclusion, there are some standards that you should adopt in order to help create more security for your system. These standards include following the ten core security principles, encrypting all information that is on a drive, in transit, and in storage. These standards also include following Triple-A to make sure that access to materials is authenticated and authorized and to make sure to perform routine tests and audits in order to help prevent attacks.  You should also make sure that your code is tested in chunks that way you can address minor issues as you build the code instead of further down the road. The last two standards include observing the DevSecOps pipeline and understanding what the consequences of waiting to implement security can do to your business and to your customers. Security should be considered constantly and implemented at every step of the way. |
| **14** | This last slide is a list of the references used to complete the project. |