# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Greetings, Everyone. Today we are discussing an overview on improvements to the Green Pace security policy. My name is Nathaniel Madore and I am the Cybersecurity Analyst tasked with providing you with this insightful overview. Let’s talk about what we have discovered here. |
| **2** | To start we have defense in depth along with this handy infographic that details the process of adding multiple layers of cyber defense to protect a system. Implementing this strategy, we provide redundancy to the security framework offering ways that if one security measure is to fail then another will prevent a complete disaster from happening. Defense in depth is an organization like Green Pace thoroughly examining the ways in which an attacker can compromise the system then taking steps to make that more difficult for a potential attacker to achieve. |
| **3** | The threat matrix is used to label examples of coding standards that should be followed to mitigate the possibility of an exploit or vulnerability. It labels the threats with priorities making it easier for the team to understand which threats require more immediate action. Next we will go on to discuss some of these principles involved in developing a threat matrix. |
| **4** | As you can see here, we have 10 principles of secure coding that Green Pace should be basing the future of its security policy around. Many of these are fundamental principles like adhering to least privilege standards for authorization or sanitizing data. These are all steps that take measures to protect data in motion, at rest or in use. As you can see from this list, we have a good basis to begin a secure coding standard for the organization to follow and to base policy from. |
| **5** | On slide 5 we are displaying some examples of what causes these vulnerabilities or exploits. Just to look at a few of these, first on the list is not to cast to an out-of-range enumeration. If casting to an out-of-range enumeration occurs, it will result in undefined behavior making the program vulnerable to exploit or to crash. Another mentioned is to not store an already-owned pointer value in an unrelated smart pointer. This is an example of a double free vulnerability where both pointers are destroyed including the one the original pointer managed. |
| **6** | Our next slide is about encrypting data in its different states. First, we have data at rest which should remain encrypted when not in use to prevent attackers. Data in motion needs SSL or VPN resources to ensure that third parties are not able to snoop or modify transmissions. Data in use encryption refers to protecting data while we use it which involves encrypting RAM. |
| **7** | When it comes to adopting a Triple A policy it is straightforward. The system should be monitored and only allowed users should have access to only the exact resources they need. It really is as simple as using logins, restricting users to necessary data and monitoring all usage of the network for accounting purposes. |
| **8-9** | Next let us talk about unit testing for a moment. It is important to test early and often. Unit testing is a way of testing code as functionality is built out to ensure that testing occurs to find as many vulnerabilities as possible along with ways. Here I have provided a few examples…(recap using slide text) |
| **10** | Use automation. Automation is a friend to the security of the organization. Use it as much as possible. This really is not an option because automation is a way of making sense of all the accounting that may take place in monitoring a system while also making sure that tests are conducted on a regular basis to discover new flaws. |
| **11** | Why is DevSecOps important and what makes it different than a normal pipeline? DevSecOps is there to embed security into the development process which in the long term saves time, money and reduces security incidents. In fact, there are many tools out that can be used for testing. Here a few just to name. |
| **12** | A common question is what are the risks and benefits?  Security is no longer an afterthought. The risk is that not testing or using security approaches to software development means that the risk of a project’s failure is much higher than without. Not to mention much more costly.  The benefits outweigh the risk four-fold. It may take more steps in the software development cycle but in the long term this will be rewarded with less vulnerabilities with the hope of avoiding a potentially catastrophic incident. |
| **13** | My biggest recommendation is to implement unit testing and secure coding as a standard into Green Pace’s policy. Layered defense will not build itself and with as many benefits as there is to it. Green pace needs to assess how secure its development is with the cautions from this report. Since security is not an afterthought. It cannot be avoided so implementing this according to best practices will ultimately ensure the success of the company. |
| **14** | In conclusion today, we have covered many topics gracing the security state found here at Green Pace. I hope that all were able to gain some insightful knowledge about what was discovered here and how to improve these processes while strengthening the information systems that Green Pace critically relies on. |
| **15** | [Insert text.] |