# CS 405 Project Two Script Template Eric Wallace Apr. 9, 2023 https://youtu.be/ZmmCb80nVTE

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hi, my name is Eric Wallace, welcome to this presentation where I will propose new security policies that will make not only software developed by Green Pace more secure but also software used by Green Pace more secure. Without further ado, let’s dive right in. |
| **2** | Here we can see a Defense in Depth illustration, the security policies introduced in this presentation is one that relies on the following of security standards and guidelines to create a multi-layered approach to secure coding. This approach is in line with Defense in Depth strategies and proposes the most security which in today’s world is needed. |
| **3** | This slide shows a threat matrix, in the C++ SEI Cert coding standards each rule has a risk assessment which contains the possible consequences that developers can use to prioritize vulnerabilities and bugs in code. They can use this prioritization to address and resolve the given vulnerabilities or bugs related to a violated rule.  The matrix on this slide shows common likelihood and severities shown on the rules, the upper left is “likely” which is a high probability if violated can lead to an exploit. To the right of that is the priority of “likely” which is the highest severity and indicates serious problems. The bottom left shows a “low” priority which is the lowest severity, unlikely to happen and has the highest repair cost. Right of that we can see “unlikely” likelihood which is a flaw with low probability of exploitation. |
| **4** | On this slide we can see the 10 principles to which the standards and practices is used by policies in this presentation. Following these practices will lower the likelihood of an attack and provide the most security to Green Pace and its customers. |
| **5** | Here are the 10 coding standards or rules used in the security policies shown in this presentation. They are listed in order by severity with the highest severity being first, for those with the same severity, they are further ordered by likelihood which is used to show the probability if ignored or violated an exploitation will take place. |
| **6** | This slide shows us the three encryption policies that if followed will ensure that even if data is somehow accessed or intercepted will prove to be useless to the would-be attackers. By encrypting data at rest, -in flight, -in use, security is increased exponentially and while it should not be the only policy practiced in a multi-layered security policy goes a long way to protecting data. |
| **7** | Triple-A policies goes hand-in-hand with the previous slide, by providing authentication, authorization and accounting; Green Pace can control who has access and the ability to edit encrypted data.  Requiring authentication by which users are to present credentials to ensure they are whom they say they are the first step, then once authentication is confirmed authorization limits or restricts users from only accessing data to which their job role allows and finally accounting, which keeps track of data a user accesses. Should a data breach take place accounting can be used to find which user introduced this breach. |
| **8** | The next four slides show the importance of unit testing on code. The four-unit tests provided in this presentation show common flaws in code introduced by developers which can be exploited by hackers. Each slide shows the result of each test has passed.  Size Less Than Number - This slide does more than the initial slide itself shows. It is indicative of how a test can be performed how inserting data into a collection can be tested to ensure things such as data used in said collection is of the data type or values that are expected. |
| **9** | Out of Range – This test checks for values that are outside the range of a collection of data. This test expects there to be an error as the value introduced by the test is outside the range of the collection. |
| **10** | Resize Length Error – This test shows how resizing a vector or collection that is larger than the max size can cause an error. This is important because when hackers try to insert data through various methods by resizing a collection or vector would result in an error thus preventing the resize from happening. |
| **11** | Can Add To Empty Vector – This is more of a verification test than anything, this slide verifies that a empty vector can be added to. Sometimes in code developers will introduce empty vectors and this test shows how empty vectors can be added to. |
| **12** | This slide shows an illustration shows automation and the stages at which automation is carried out. This can be important because when developing software following this summary can guide developers of when they are to implement automation into the process of making code more secure. |
| **13** | The stages at which Green Pace will modify their current DevOps process in the “Verify and test” and “Design” stages.  The implementation of new static analysis tools to scan code for vulnerabilities, test code, detect dead code, and errors in code according to standards and guidelines.  The code will need to be modified to ensure the code follows current standards and code.  Each time code is modified, static code analysis will need to be performed to determine if the newly developed code introduces any additional vulnerabilities or error in code.  Additionally, penetration testing can be carried out during transition and health check stages to make sure the code does not allow unauthorized access to it via the internet.  By following the recommendations above security will be greatly increased and reduce the likelihood of releasing software with vulnerabilities and bugs. |
| **14** | Deciding whether to act on a potential threat cannot be understated. The risk of procrastinating to address a potential threat can be magnified by the likelihood an attack may take place and the severity if an attach should happen. Should an attack take place the result could be loss of time, damages to equipment, increased cost, and reduction in Green Pace’s reputation. Releasing software or developing code with known security risks can be catastrophic.  Implementing security as code is developed can prove to be very beneficial, it can also reduce the amount of time to produce a working product and a software that is much more secure. |
| **15** | It is my recommendation that Green Pace provide to its customers a mechanism for updating software developed for them, so they may receive updates to software to resolve vulnerabilities or exploits found after deployment so attacks may be prevented.  Another recommendation is the education of developers in the latest security standards and techniques used by hackers to gain access to data being held, transmitted, or used by software. This could reduce gaps in security and result in a more secure product. |
| **16** | In conclusion, having a plan for if or when an attack takes place is vital. Having security policies and contingencies are paramount in today’s world. No matter how hard we try, it is impossible to account for every scenario and be 100% protected against threats, following standards and guidelines shown in the previous slides can provide the most protection.  If developing software for Green Pace customers, it is very important they can update their software to reduce the likelihood of an attack. The last thing Green Pace wants is their reputation to be tarnished by the releasing of software which is tied to a data breach or leakage.  Thank you for attending this presentation. |