# CS 405 Project Two Script Template

Colton Thompson - Project Two: Security Policy Presentation

April 10th, 2021

YouTube Link: <https://www.youtube.com/watch?v=ZyTNG3MzpM0>

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hi there, my name is Colton Thompson, and this is the security policy presentation for Green Pace. |
| **2** | Security is important in software development and to protect Green Pace from the wide variety of attacks that exist as well as new attacks being created every day this presentation will help Green Pace to prevent and protect from would be attackers.  This policy will be based off the Defense-in-Depth diagram shown in this slide. |
| **3** | This slide represents what I believe to be the security risks that Green Pace currently face and should prepare for using my security policy.  Data breach – It is possible that if an attacker were skilled enough, they could potentially break into our network and steal information of our entrepreneurs and other clients. This type of attack could occur if we are slow to update system software as well as normal operating system updates and patches. All logs and monitoring services need to be examined to insure we are always staying on top of security. If an intrusion is detected, it needs to be handled immediately.  SQL Injection – All input related systems need to be protected from SQL injection from always true conditions to prevent unauthorized access to our networks and systems. This can easily be fixed by enhancing our code reviews and unit testing procedures and practices.  Buffer Overflow – Our servers need to undergo the same level of code testing and code reviews to ensure errors such as buffer overflows do not occur and cause our systems to crash which could cause further security concerns such as data breaches.  Source Code Leak – To prevent any sort of leak of data or source code leak all our data should follow the encryption policy described later in this presentation. The more boundaries we can create to slow down the attackers the higher the chance of mitigating attacks before they can cause real damage to Green Pace. |
| **4** | These are the 10 principles of security we need to follow to keep Green Pace safe as developers.  Validating all input data - means that when a user types something into our system we need to run that data against filters to prevent various syntax being used against us such as what you would see with an SQL injection style attack.  Heed Compiler Warnings means to always pay attention to warnings we receive from the compiler during development. Any warning can turn into a much larger error or crash that could lead to a breach in our security. It is important to follow safe coding practices to keep Green Pace from experiencing an attack.  Architect and Design for Security Policies means that before code is written each type of threat should be assessed and prepared for with our design. If we implement a design that can ward off an attack this makes our systems and software more reliable and more stable.  Keep It Simple means that sometimes solutions are simple and should not be overdone. Complexity can cause slowness when it comes to security especially if the people who originally created the code are no longer employed at Green Pace and needs fixing, the developers now may not be able to acclimate fast enough to solve the problem to prevent an attack.  Default Deny means as the default level of access we should always deny rather than permit someone into our system or network. No access as a default protects us from would be attackers attempting to get in easily.  Adhere to the Principle of Least Privilege means that we limit everyone’s access to what is necessary for their job function or task at hand. No excess is given to compartmentalize data from being accessed by the wrong person if it does become breached.  Sanitize data sent to other systems meaning that anytime data needs to be transmitted to another system either over the network or local it needs to be cleaned of any special characters or syntax that could be used to cause a crash or errors on our system that opens an opportunity for attack.  Practice Defense In Depth means to follow the security mission in the defense in depth diagram. Defense in depth represents the many layers of security that should be employed by all software development companies like Green Pace, and it acts as a guideline to keep all software, data and clients safe from attack. |
| **5** | These are the coding standards that are listed in my security policy that I believe are important standards to follow during the development of our software.  SQL Injection should always be prevented and the best way of doing this is to use prepared statements when sending queries to the database. Be wary of any always true conditions being sent through an input field to test our security.  Exception safety is important because missing an exception that gets thrown can mean a potential crash of our software and could mean further security risks such as a breach if it is not handled properly.  Division by Zero operations are simple in concept but still happen, be cautious when performing arithmetic operations to ensure we are never dividing by zero and always catching it before it happens.  Buffer Overflow is another huge problem in software development and happens frequently. It is important to make sure we are following secure coding practices with buffers and how we use them as they are a common cause of instability in systems.  Memory Protection is important and is listed on this slide multiple times because of this importance. Accessing Freed Memory can happen due to overlooking code and should be caught in code reviews and code analysis. Reading uninitialized memory can cause crashes and leads to instability.  Dereferencing Null Pointers is a common issue in software development and is important that we keep track of all pointers when we reference them. These issues can easily be solved by using code analysis software and code reviews.  Selection statements, String Correctness and Assertions all follow a similar pattern as above, follow secure coding practices and principles and make sure to perform code reviews and run code through code analysis before each stage of development proceeds. |
| **6** | There are three pillars of encryption that we will follow in my security policy for Green pace.  Encryption In Rest means that all data is encrypted when it is in storage and not being used.  Encryption In Flight means that all data is encrypted before and during transmission to a new location whether local or otherwise.  Encryption in Use means that all data is encrypted while in active use such as in random access memory on a system.  The general theme of this policy is to ensure that all data is encrypted when possible to prevent attackers from accessing the data. |
| **7** | This slide is for my Triple-A Policies for Green Pace  The three triple A policies are Authentication, Authorization, and accounting.  For authentication, each user of the system will need to provide proper authentication to be a valid user of the system. This means that all users need to go through a login process and perhaps a second level of authentication to validate that they are who they say they are.  For authorization, each user needs to be authorized for each area of our systems and software to prevent users from accessing something the otherwise should not be accessing. For example, a code quality tester should not have access to user accounting information.  For accounting, all user accounts will be logged by recording all actions that they perform such as what changes they make, what files they access, what new user accounts they create as well as what time they happened. |
| **8** | This is the DevSecOps diagram that will be discussed in the next slide. |
| **9** | The DevSecOps diagram that was shown on the previous slide demonstrates the various tools at our disposal to improve the security for Green Pace.  DevSecOps is a pipeline that involves two stage of software development which are Pre-production and production.  Each stage is important for building security now rather than adding it later. If security is built too late it can be far less effective especially if it does not easily fit with the previous design.  During pre-production, the cycle begins with assessing current threats that the software could face now and in the future. This assessment is ongoing even during production to mitigate new threats as well.  Once the assessment stage is complete the cycle moves on into security driven design which takes the information gained in the assessment stage and puts it to work in terms of defending against those threats from a design level. After the design phase the software the software is built using trusted repositories and secure open-source storage. After the build process, the build goes through a verification and testing phase. During this testing phase the build goes through a series of tests such as vulnerability, functionality, as well as compliancy and security.  This completes the pre-production part of the DevSecOps pipeline.  During the production cycle of the DevSecOps pipeline a few things change.  The production cycle begins with a transition and health check where we configure and deploy the software with security settings and a round of penetration testing based off the assessment phase done earlier during pre-production. The next step which repeats is to monitor and detect any potential threat to the system. The monitoring is done through log collection, SIEM analytics, event alerting and intrusion detection. If an attack is detected, we respond by blocking the attack, turning off services and/or rolling back data & systems. From this point we work to return to a security baseline or a stable state after an attack. |
| **10** | By continuously monitoring everything that interacts with Green Pace’s systems and network we can identify the potential threats and make more informed actions such as the attack responses shown in the DevSecOps pipeline.  If we act too quickly without the proper information, we could make mistakes that result in further damage to Green Pace. The same can be said if we act too slowly that an attacker manages to successfully attack us and steal user information or even source code. |
| **11** | My recommendations are to remember that there is no such thing as a perfect defense.  Staying vigilant to potential threats and attackers and following the security guidelines provided by Defense In Depth in conjunction with the guidelines listed here in this presentation. |
| **12** | In conclusion, I believe that my security policy displayed here in this presentation will aid Green Pace in its security program to protect its assets and its clients. |
| **13** | [Insert text.] |
| **14** | [Insert text.] |