# CS 405 Project Two Script Template

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

|  |  |
| --- | --- |
| **Slide Number** | **Narrative** |
| **1** | This is a presentation of the Green Pace security policy. This policy is meant to provide guidelines for the implementation of recommended security features. |
| **2** | Defense in Depth (DiD) refers to a series of layered defenses of a computer system or network. This series of varying mechanisms can include firewalls, network segmentation, and strong passwords. Many other mechanisms can be used for this defense strategy but the key concept with DiD is that the mechanisms for defense are layered in a series with the goal of protecting the confidentiality, integrity, and availability of the and the network or data |
| **3** | Validate Input Data  When a user enters information into the software it will need to be validated. This ensures that the software is getting the information that it is expecting. This can prevent crashes and bugs. It also prevents malicious actors from entering information that corrupts or grants unwanted access to a database or server.  Heed Compiler Warnings  Compiler warnings are added to software because they help fix known issues. Many times these compiler warnings can prevent bugs from happening later on in development but can also help secure code. Typically if the compiler is giving a warning it is flagging a known issue which the developer should interpret as a security issue or bug that has already happened and can be prevented.  Architect and Design for Security Policies  In order to protect the integrity of the software system, developers should examine how the safeguards are being implemented in their software. This implementation of the software safeguards and enforcement of the security policies can help to protect the information being handled by the system. This will protect how data is used, stored, and retrieved.  Keep It Simple  Some of the most basic solutions are the most effective. Keeping the design of the software simple also makes it easier to secure the software. If all the parts of the software system are easily understood then the development team will be able to employ much more effective security strategies.  Default Deny  The default mode of the software should deny access. Only users that have been authenticated and identified should have the ability to access part of the system or software.  Adhere to the Principle of Least Privilege  One part of the program may not require information from other parts. It is important to limit the access to information to only what is needed to accomplish the task of that portion of the software. This will reduce the risk that hackers can gain access to critical parts of the system from an unexpected source.  Sanitize Data Sent to Other Systems  It is important to specify the data that is being sent from one system to another. If a function is expecting an integer and it receives a string then errors can occur. It is important for the system to check the data that is being passed to ensure that it meets the expectations of the parts of the system that will be consuming the data.  Practice Defense in Depth  By layering the defense strategies the team can increase the efficacy of the different security strategies. This will help the team detect attempted attacks and also makes it much harder for malicious actors to gain access to critical parts of the software system.  Use Effective Quality Assurance Techniques  Quality assurance ensures that the software being produced meets the standards of the company. These techniques should be incorporated into the development process to ensure that the final product is secure and meets the needs of the stakeholders.  Adopt a Secure Coding Standard  Unsafe coding practices can result in the loss of information. This can also damage the reputation of the company. It is important for all team members to adopt secure coding standards as part of their normal repertoire. |
| **4** | For the coding standards we selected the ones developed by Carnegie Mellon University. This site supports the development of coding standards for commonly used programming languages such as C, C++, Java, and Perl, and the Android platform. These standards are developed through a broad-based community effort by members of the software development and software security communities. |
| **5** | Encryption in rest  Ensures that data stored on disk is encrypted which makes it more difficult for a hacker to access the data in the event of theft.  Encryption at flight  This is the process of encrypting data while it is being transmitted.  Encryption in use  A method for using data while it is in its encrypted state so that decryption is not necessary for an application to run. This can be done by trusting the hardware or trusting the end application. |
| **6** | Authentication  The process is verifying the identity of the user with user logins via password or two-factor authentication.  Authorization  The process of determining the user’s level of access. Specify what an authenticated user is allowed to change on a database and how often they can access the database. The addition of new users should be a limited privilege granted only to administrative users.  Accounting  The process of keeping records of all user actions in logs. This should include all of the files accessed by the user. |
| **7** | The coding vulnerability that was tested was the collections vector. Tests for this vector were meant to test the input as well as changes made to the vector. The vector was tested at creation and tear down also. |
| **8** | Pre-commit checks consist of steps to complete before the developer checks code into the source code repository.  Commit-time checks are automatically triggered by a check-in to a source code repository.  Build-time checks are automatically triggered by successful commit-time checks.  Test-time checks are automatically triggered by successful build-time checks.  Deploy-time checks are triggered when additions to the repository have meant all of the above criteria and the team can perform additional in person checks such as peer review or code walkthroughs. |
| **9** | The risks and benefits of enacting the system increase the sooner the measures are taken. The risks of not enacting a security plan could include a loss of data or intellectual property. Or damage to the companies reputation. |
| **10** | Areas for improvement would be unit tests for all data transferred in the system. Another improvement could be the encryption of data as it is passed throught the server. The gaps in the security plan are that it is not scaled over time. New threats are not accounted for. The best way to resolve these issues would be to create a team with the goal of developing and securing code as it is created. |
| **11** | Building your devsecops PIPELINE: 5 essential activities: Synopsys. (2019, September 18). Retrieved April 16, 2021, from https://www.synopsys.com/blogs/software-security/devsecops-pipeline-checklist/ |
| **12** |  |
| **13** |  |
| **14** |  |