# CS 405 Project Two Policy Presentation

Valerie J. Smith

June 17, 2023

YouTube link: https://youtu.be/JKFD7br\_AVg

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello, everyone, my name is Valerie Smith, and I am a developer for Green Pace technology. The presentation today will cover the software security policy that Green Pace has required to protect our software and data. |
| **2** | The security policy prevented will assure that our software design processes and build tools are incorporating a Defense in Depth approach, which will help to secure our assets in layers of security countermeasures. |
| **3** | The threats matrix table shown has been created by analyzing the chosen code stack for common coding errors that have previously resulted in known security vulnerabilities. The table reflects the potential issue with the code stack, and the probability of a security vulnerability. The potential vulnerabilities were found by using the examples from the Software Engineering Institute wiki by Carnegie Mellon University. The potential security vulnerabilities can also be found in code during the development and build parts of the CI/CD pipeline by the use of automated security scanning tools. |
| **4** | 10 principals of software engineering have been selected to ensure that security is incorporated into every step of the SDLC lifecycle, creating a DevSecOps environment for Green Pace. Noticeably, the addition of the principal of Defense in Depth has been added to this list. By combining some newer standards with some basic time tested security guidelines, it will be possible to architect and design for security in order to achieve a secure environment. |
| **5** | For the Green Pace tech stack, the following coding standards have been analyzed to create a secure coding development environment. The table shows the rule and the likelihood of the potential for a security vulnerability. The severity describes the impact to the business if a security vulnerability is found based on each rule. The priority column orders the priorities by value, for example, the higher the value, the greater the priority for a particular rule. |
| **6** | Encryption policies for Green Pace cover all aspects of data at any stage. Data should be encrypted while it is at rest, such as while being stored or not in use. Data should be encrypted in flight, such as when traveling or being transmitted, and protected with HTTPS, and SSL/TLS protocols. Data in use, such as sensitive data as credit card numbers and passwords, should be encrypted the entire time that they are in use so that they are never left unsecured. |
| **7** | Triple-A policies are defined to discover who is accessing the system and what they are allowed to do while they are there. Authentication will identify a user and ensure that they are who they claim to be. Authorization describes what duties the user is allowed to perform with the use of a role-based access system. Accounting will monitor what resources were accessed, who accessed them, what commands were issued, and what data was sent and received. |
| **8** | The unit tests shown are examples of testing that is done at the development level as a part of secure coding. These unit tests test the usage of the memory in a collection. As memory is an area where many security vulnerabilities are possible, it is important to ensure that memory management is tested thoroughly. The unit tests were created using the Google Test framework. This test in particular will show that an out of range exception is expected when a collection is given data that is too large for its size. The test detail summary shows that the test was successful. |
| **9** | This unit test is testing a popular method that alters the size of a collection by removing one element. The test shows that the method removes the element and that the original size is no longer true. A negative test like this one will show that the original statement is now false. The Test Detail Summary shows that the test is successful. |
| **10** | This unit test is testing another popular method to change the size of elements in a collection, which is a critical factor in managing memory in an application. The Test Detail summary shows that the test is successful. |
| **11** | This unit test is also testing the same popular method from the previous slide to change the size of elements in a collection but is implemented using a negative test. The negative test asserts that the original value is not the same. The Test Detail Summary shows that the test is successful. |
| **12** | The Automation Summary shows the entire flow of the DevSecOps pipeline. In this image, the areas where security is incorporated are shown in every step of the process. |
| **13** | Tools that are implemented in the DevSecOps pipeline provide security for both application and infrastructure while the code is being written and as it is being built and deployed. Automation is an important part of the DevSecOps pipeline and covers unit testing and as well as code scanning tools that are scanning for common known security vulnerabilities. Monitoring tools are used to monitor uptimes, identify anomalies, and discover malicious activities. |
| **14** | There are risks and benefits in DevSecOps as compared to former DevOps methodologies where security was often excluded until the end of the application development or not included at all. As today’s applications are becoming increasingly complex, security can no longer be kept separate or discluded. Benefits of adding security in DevSecOps include preventing threats by the use of access control, data classification, encryption, remote access, acceptable use, patching, malicious code protections, physical security, and more. |
| **15** | In addition to implementing this security policy, it is also important to ensure that application developers and engineers are participating in continuous education to keep their skill set up to date and to learn about recent security vulnerability issues. Developers that engage in these activities will sharpen their critical thinking skills, saving time and incorporating security from the start. Agile methodologies are recommended as a part of the DevSecOps development to garner communication and collaboration among developers and security professionals in today’s rapid development environment. |
| **16** | In order to conclude this presentation, a look to the future of cybersecurity is necessary in order to understand why the need for a security policy exists. In addition to maintaining and updating the policy, it is important to understand that securing networks, data, devices, and identities will be continued in the future of cybersecurity. A business needs to have the ability to recover when they are targeted, and new frameworks such as Zero Trust are emerging. Ransomware and other attacks will continue and become worse. Mobile device security is becoming more important with the advent of remote work, and addressing security and automation is prevalent due to workforce skills gaps or shortages. A push for cybersecurity as a service is underway. |
| **17** | I wish to thank the following references for their knowledge and expertise as a guide in the creation of this presentation. Thank you all for watching. |