CS-405 Secure Coding

Project Two: Presentation Script

Anthony Lewandowski

December 19, 2021

**Slide 1**

Hello everyone, my name is Anthony Lewandowski, and I am the developer that created the new Green Pace security policy. This will be a short presentation on principles, coding standards, and a few other aspects that encompass secure coding.

**Slide 2 – Overview**

I am introducing the new Green Pace Security policy for a safer, more secure work environment. The primary purpose of this security policy is to create basic rules and procedures for all individuals working in the development environment. Following the rules and procedures in this security policy will help ensure that everyone at Green Pace is doing their part to support secure coding. Below we can see a visual overview of the concept of Defense in Depth.

**Slide 3 – Threats Matrix**

Here we have a threats matrix. The likely and unlikely coding standards are listed in the diagram's top left and bottom right. In the bottom left and the top right, we can see what low priority and priority mean within the security policy's standards area.

**Slide 4 – 10 Principles**

Here we can see the ten principles that I have applied to the Security policy. These are the core security principles for writing C or C++ code and should be applied to any work done at Green Pace.

**Slide 5 – 10 Coding Standards**

On slide five, we have the ten coding standards directly related to the ten principles. If you scroll down to the risk assessment section in the security policy, we can see that each coding standard has a priority level. The higher the priority level is for a particular coding standard, the higher the risk level, so it must be handled as soon as possible.

**Slide 6 – Encryption Policies**

This slide presents a brief overview of how a developer should use encryption. We can see that there are three different types of encryptions, and they all serve a purpose depending on the state of the data.

**Slide 7 – Triple -A Policies**

On this slide, we can see a brief overview of the triple-a policies. Authentication is identifying a user with a username and password. This is generally the first line of defense against unwanted users.

After authentication, we have authorization checks for a users permissions within a system. Authorization will provide users with access to sections of a web service. This is typically another security layer to keep unwanted users out of the system.

Finally, we have accounting. Accounting is responsible for monitoring and logging user activity within a system. Accounting is used to determine the type of permission to give a user and is an excellent way to track malicious activity.

**Slide 8- Unit test 1**

The following four slides will be examples of unit tests performed to eliminate code vulnerabilities using the C++ language. In this test, we verify the max size of a collection after it changes. We use the EXPECT greater than or equal assertion to compare values after adding entries. At the bottom of the slide, we see that the test has passed.

**Slide 9 – Unit test 2**

In the second unit test, we use ASSERT\_FALSE to assert that 100 entries were added to the collection. The test fails because we have successfully added 100 entries to the collection, so it is not false.

**Slide 10 – Unit Test 3**

In the next slide, we are testing to see if the collection.erase function erases the entire collection. We must create a vector and tell the function where to begin and end to erase everything. After the test runs, we can see that it passes.

**Slide 11 – Unit Test 4**

In this unit test, we are testing to verify if resizing increases the collection we are working with. We add entries to the collection, resize it, and use an assertion to confirm the collection size. After we run the test, we can see that it passes.

**Slide 12 -Automation Summary**

The best areas in which automation would be helpful would be in testing, health checks, and monitoring. These areas typically require a lot of repetitive code to be written, consuming a lot of development time. With the implementation of automation in these areas, tests could be run and completed at higher speeds than a human would be capable of doing. Development time would increase as well as the quality of software.

**Slide 13 - Tools**

The DevSecOps pipeline is a process of integrating security into the software development life cycle. The process contains security practices to build test and deploy secure software. External tools used in the pipeline may vary, but an example may be automated test frameworks. Automated test frameworks could aid in exhaustive testing and could be found in testing, health checks, and the monitoring areas of the diagram on the previous slide.

**Slide 14 – Risks and Benefits**

If a development team leaves security risks problems until the end of development, the project's outcome will not be favorable. If a development team leaves security risks until the end could end up costing a lot of money to fix. Vulnerabilities in code can multiply, extending the project's deadline.

A better approach would be to develop software with security in mind. In this mindset, security is considered throughout each step of the development phase. The benefits include helping reduce the cost of a project by reducing code vulnerabilities leaving a more secure product.

**Slide 15 – Recommendations**

In this slide, I mention Green Pace did not cover two potential gaps in this security policy. The first gap would be a defined process for dealing with third-party vendor risks. A second would be a well-defined procedure for dealing with incident handling.

**Slide 16 – Conclusion**

To build on what I had mentioned in the previous slide, a structured procedure for dealing with incidents (such as malicious attacks) before they occur would be highly advantageous to preventing problems that might arise at the moment. If people are educated on a particular plan, the success rate in that situation would significantly increase.

Procedures for investigating third-party vendors would also be an excellent standard to ensure that sensitive company information is secure. It is essential to know who you are working with and if they have a high level of security. Creating a procedure to investigate third-party vendors would be an excellent decision to ensure company data is in good hands.