# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello Viewers, My name is Francis Ugorji. This is a video presentation Green of pace’s secure development policy that I developed for the Green Pace Company, it’s a summary of security policy and coding standards, to be followed strictly by the Green pace company’s employees. Green pace company is a company that has committed to better coding standards and implementing strict security polices |
| **2** | The main purpose of the security policy is to have the company moving towards the right direction when its handling software and data. This method is known as defense in depth strategy, as illustrated in the diagram. As we can see in the diagram, there are layers of security existing, with the primary role of deepening the layers of protection the company has against threats, this process consist of layers that starts off with the physical security existing for protecting data and software like CCTV, access control to buildings, alarms etc. Then Comes server level protection which is termed cloud security, where the computer hosting data and software is protected from network intrusions that could occur through the internet, even through ISP providing the internet connection.  A good practice for such a protection would be to use wired connections for example, to protect the system and software, that could happen in house and onsite where internet is reaching the server or system. The layer of security continues and deepens as the software and data leaves its comfort zone to its respective authorized users. The layer continues to perimeter security like using Network security like Netflow, Host security like Encryption, endpoint security like Man in the middle attacks, App security like patch management as well. Governance and Soc practices are also put into account when it comes to fulfilling the mission of setting up a good security practice.  To illustrate with an example that was given in module 3 discussion, the attacks that were listed in my discussion board response listed 3 different types of attackers, white hackers, black hackers, and grey hackers. White hackers as someone would guess are hackers with good intentions that find vulnerabilities in system, software or network and present them to the appropriate bodies or companies, while black hackers set out to take advantage of such vulnerabilities and hack for gain or damage to potential targets. Sometimes, the sell proceeds from their attacks like passwords , login information, etc. Grey hackers on the other hand find vulnerabilities and then go ahead and fix them, and sometimes alert the owners of such systems or software of the attack and the applied fix that was administered. In my discussion 3 examples that I gave, three attacks were listed, the code red worm attack that affected IIS web server that was identified by a white hacker Marc maiffet, that attack when compared to the diagram of defense in depth would come under APP security or cloud security for those leasing web server from a third party. The mikrotil router attack that was found and fixed by the grey attacker Alexey, would fall under network or perimeter security. Routers have a lot of power when it comes to defending a system from intrusions and in this case, the router failed to do so. The last attack that was listed was by black hackers that attacked software (the orion software), and inserted malicious code into it. This particular instance falls under App security. |
| **3** | The table below represents the threat matrix of the security standards created. They are grouped based on their priority and severity in terms of the level of necessity to implement them. As we can see, number 7-10 can be seen to have a high priority and probability, while number 1 has a very low severity and probability |
| **4** | In this slide, we can see the 10 principles being matched with standards that fit the principle in question. The principles range from validating input data, to sanitizing data sent to other systems. Some of the principles prefer keeping codes as simple as possible for simplicity sake, while others want as many compiler errors as possible handled prior to the error reaching the compiler. |
| **5** | In this slide, the different coding standards are listed with a summarized explanation of their functionality or how they are able to minimize threat and secure code. Some of these standards want the use of assertions to test out code prior to deployment as we can see in the case of STD-006-CPP, while others  like encouraging deallocation of resources after usage to prevent memory leaks as it is in the case of STD-005-CPP. |
| **6** | Encryption represents a standard method used for securing data, as it goes from one place to another, to avoid it being readily available to unauthorized users. In this slide, the three states of encryption of data are explained. The first state which is encryption at rest represents when data has been encrypted and stored. This is the case of encryption of data in a devices like USB drive, network drive, and encrypted data as in the case of locked PDF files, locked zip files, etc. The second state known as encryption at flight represents when data is kept in encryption state as it travels from one place to another. The last state being encryption in use represents keeping data encrypted as it is used by the system. |
| **7** | This slide explains the triple A policy of Authentication, Authorization, and Accounting. Authentication goes into action when we try verifying the identity of a user, authorization is when we grant the user access to specific files that they are authorized to have access to, without exposing other files that they are not authorized to have, while accounting Represents record keeping and transaction records of activities by users |
| **8** | The next six slides are slides of unit tests that were conducted during project 1. The first one here as we can see is resizeincreasrcollectionsise. This testing is done to confirm the accuracy of written code when it comes to the method as shown. The method here checks first to see if collection is empty, then checks the size before subsequently resizing the collection. Afterwards, it checks to see if the collection was successfully resized. |
| **9** | The Unit testing does something similar to the one in the previous slide. This test is setup to check if the code will throw an exception when a larger size is requested than the required size. This type of test is called IncreaseCOllectionReseverAboveMaxSize |
| **10** | In this slide, Entries are added, and a test is run to check if the proper size exist |
| **11** | In this slide, the collection size is increased using a parameterized method, and the size of the collection is checked after addition. |
| **12** | This slide demonstrates testing for out of range exception, which is expected to be thrown when a non existing entry is requested by a user |
| **13** | The unit test in this slide tests to make sure that the target collection has been successfully cleared after a user requests for it to be cleared and it attempts to clear it |
| **14** | The diagram shown represents the cycle of devsecops Automation, from pre-production to production. There are about 8 different steps suggested in the diagram, from assessing and planning against threats, to having a plan for responding to attacks like blocking attacks. The devsecops workflow encourages automation. Through automation, the development life cycle of software is enhanced. The process of planning and setting up security of software should be highly automated and straightforward for it to be successful |
| **15** | As we saw in the devsecops diagram, there were two stages, the pre-production and production. In the pre-production stage, the planning, design, building and testing of software is done, while in the production stage is when the software is deployed, monitored after deployment, maintained and checked for errors and problems. |
| **16** | One may wonder what the benefits would be to implement a security measure at the early stage of software development. Such a step has its advantages and disadvantages. One advantage is that its much easier to implement a security measure in the beginning of a project than to include it at the end. |
| **17** | Some of the disadvantages is that it might slow down the development process, and it might force the company to spend more to train developers, but then acting later might raise the cost of adding security measures later. |
| **18** | One very important recommendation is to make sure there is full compliance to security measures being put in place. The required training needs to be made available to developers, and the whole development life cycle should have security implemented in all aspects of it. |
| **19** | In conclusion, its very important to have a security plan for your software, which should be implemented. Training of developers on security policy and standards is necessary, and enforcement of security measures should be made a top priority. |
| **20** | Here are three references for this PowerPoint, thank you and have a nice day |