# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello! my name is Michael Neff I am currently enrolled in my last  semester of college for my four year degree in computer science.  This is my security policy presentation. |
| **2** | Defence in depth can be viewed as a securty mechansim that has multiple layers of defence in order provide defence and protect system vulnerabilities. |
| **3** | Threats Matrixs. This slide shows an easy to read chart that boosts my personal choices of the rules of each princple. I labeled the rules and fit them in the corresponding boxs. For example F1051-cpp could be sceen medium threat and unlikly to happen. It shows more simple attributes about the Rule. |
| **4** | This slide show the 10 principles. Under each principle we can see that I placed a Rule. The rules are specifically allocated to fit the profile of each rule, and these principles define the roles that the rules fall under. |
| **5** | The coding standards listed are columned by many factors. We simply take the rule, define the severity of the rule, if its high, medium, or low. And then follow up with the likelyhood of seeing the rule. Understanding the remedition cost, and priotity of the rule is important as well. As you can see for example we have the rule DCL12-C with a low severity, and showing that it will unlikely arise. When it does this boosts high costs. |
| **6** | Encryption In rest Is data that is stored on a physical memory device, like hard drives, SSD’s and even as far as cloud storage. The encryption at rest protects data from an unauthorized breach. It plays a crucial role for defending sensitive information from exposure, especially when a device becomes lost.  Encryption in flight Data that is being transferred from one location over a network used encryption in flight. The end game is that encryption in flight is protecting data against interception while being transmitted.  Encryption in use Is unlike encryption at rest and flight. This encryption protects data specific to states, and is used to secure data while in use. This protects against threats such as unauthorized or non authenticated breaches, and or memory scraping. This form of encryption ensures that information remains protected. |
| **7** | Authentication is verification of identify of users or systems before access is granted. There are many ways to authenticate a user. Unique passwords, biometric and pin codes, multi factor authentication is also common. These steps ensure that the information is correct and can provide access to anyone who is authenticate.  Authorization is the process after the authentication verifies identity. Authorization defines the levels of permissions granted to a user. This can be done to specific roles if relating to a work place system.  accounting.  Accounting involves tracking and recording of users data. This includes logging, file access, and changes to usage. This is essential for monitoring security compliance and it also helps ensure that policies and security’s are in place. |
| **8** | Testing can be done to ensure that proper buffers and overflow errors can be detected. Tests like ensuring collection size if its -> if its empty or collection size not empty. many tests can be performed such as seen on the screen shot. |
| **9** | The automation summary diagram is a guide of Dev Sev Ops. The design phase is the phase that involves planning while keeping security in mind. Threat modeling and risk assessment and defining security polices are also done in the phase.  During the Test Phase testing is just not about functional testing but ensuring it includes securities in the pipeline. Health checks are done to ensure that functions are working as expected.  The respond phase focuses more on the immediate steps once a security threat is detected. Implementing automated responses and having a well defined response plan that implements escalation procedures.  Overall the summary diagram provides step by step milestones that allow developers to follow to create a robust program. |
| **10** | What Is DevSevOps Pipeline?  The pipeline intergrates security into DevOps workflow hence the SEC. This ensures that security is shared across the development phases. Building and testing and deploying software quickly and securely with multiple security checks throughout SDLC.  Summary Diagram  The summary diagram discussed is the key stages to DevSeVOps  As spoke before planning and design intergrates security in the beginning.  Secure code helps coding guidelines, such as ERR60-CPP rule spoke about previous which is the Sanitize Data Sent to Other Systems. IDE’s may provide real-time feedback on code quality and security.  OSWAP Threat Dragon  Oswap is a model that is used to identify potential risk in the early stages of design.  CheckMarx  Is a tool that is integrated that scans the code for any vulnerabilities during the build phase.  Container Tools  Containerizing tools is a very interesting topic to arise during this process im currently taking CS470 full stack 2 and during deployment stages in AWS cloud containerization was a huge process in our final assignment. Using these tools during deployment ensures that the configurations are consistent and applied across the board. |
| **11** | Problems  Security is often an after thought. This leads to vulnerabilities being discovered toward end game development. Often delaying deployment, and costly fixes, comes with reputation damage if not found before releases.  Solutions  Intergrating security tools into exisiting DevOps pipeline early can alternatively  Reduce problems, and create early detection and reduce code cost, and reputation status  Risks  Initial higher costs with a long term saving and faster remediation eventually out way the lack of implementing DevSecOps practices and tools in early on build phases. Careful planning minimizes disruption and ensures a smooth transition into all phases.  Steps To Take  Evaluation is huge, and evaluating the current status of the pipeline to identify vulnerabilities is a must take first step. |
| **12** | Gaps like not addressing phases of SDLC in early stages can impact final deployment, vulnerabilities could be introduced early and remain undetected until late stage such as plan and design.  Security polices may not have compliance measures to monitor applications after deployment, meaning threats could go undetected and breaches can happen in the future.  Outdated management is another gap. The polices in place many not address regularly mainaning and updating and patching software and libraries that are exploited by attacker. Including a robust management process ensuring patching and updates are being done is a huge part to ensuring proper code management and a big part of the SDLC process. |
| **13** | Applying good coding practices and following a well-structured plan are crucial for successful deployment in DevOps. Integrating security into DevOps (DevSecOps) not only reduces costs but also helps ensure an application is free from vulnerabilities. Understanding and monitoring risks, as well as their frequency, are essential for preventing breaches and maintaining a secure program. |
| **14** | [Insert text.] |