# CS 405 Project Two Script Template

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

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | * “Defense in depth” is a common security practice that involves layering security controls to protect an organization’s assets. Independent layers of security controls are used so that if one control fails, the anomalous activity does not go unchecked. In short, defense in depth is about eliminating blind spots and holes in your defensive security strategy. * Here we can see the defense in depth illustration of all of the layers. Physical security, cloud, perimeter, network, host, endpoint, app security layers to protect critical assets systems and data. |
| **2** | 10 CPP standards,  Organized by severity and priority  Memory protection P18 High  Throw and catch exceptions Unlikely P4 Low |
| **3** | 1. Validate input. Validate input from all untrusted data sources. 2. Heed compiler warnings. Compile code using the highest warning level available for your compiler and eliminate warnings by modifying the code 3. Architect and design for security policies. Create a software architecture and design your software to implement and enforce security policies. 4. Keep it simple. Keep the design as simple and small as possible 5. Default deny. Base access decisions on permission rather than exclusion. 6. Adhere to the principle of least privilege. Every process should execute with the the least set of privileges necessary to complete the job. 7. Sanitize data sent to other systems. Sanitize all data passed to complex subsystems. 8. Practice defense in depth. 9. Use effective quality assurance techniques. Good quality assurance techniques can be effective in identifying and eliminating vulnerabilities. 10. Adopt a secure coding standard. Develop and/or apply a secure coding standard for your target development language and platform. |
| **4** | 1. [Data Value Priority 9 2. Data Type Priority 9 3. String Correctness 18 4. SQL Injection Prevention Priority 18 5. Memory Protection Priority 18 6. Assertions Priority Priority 1 7. Exceptions Priority Priority 4 8. API Vulnerability Assessment Priority 4 9. Secure Object Oriented Programming Priority 6 10. Expressions Priority 6 |
| **5** | Encryption at rest is data that isn’t being used on a hard drive. If the attacker gets access to the hard drive the attacker must hack the data without the key. The encryption at rest also means that the data is not travelling between the devices. Encrypted data at rest can be saved to a hard drive or portable device. Encryption at flight is data travelling from one point to another. This process encrypts data while it is being transmitted. Such as emails, messengers, calls. This type of data is usually less secure than data at rest, and it is an easy target for hackers. It is the data that is being actively used by the user with the encryption key. In this stage data is vulnerable. It can lead to a human error and expose the data. |
| **6** | Authentication is a process to verify user identity such as password and a second type of authentication can be a phone number with a text message. The most current is DUO or Microsoft multifactor authentication. User logins.  Authorization for data access determines what groups user can be in. Write, read, full access permissions groups usually are getting authorized with the administrator. User level of access. Addition of new users  Accounting is process to monitor users actions with the data. It usually keeps track of what user can or can’t access, and what changes were made to the file stream. Files accessed by users. Changes to the database. |
| **7** | 1st Unit test. Data Value and Memory Protection.  verify adding five values to collection |
| **8** | * ASSERT\_TRUE(collection->max\_size() >= 10);5;1;0 * We use this values to verify that the max size is greater than or equals to the size for 0,1,5,10 entries |
| **9** | * int previousSize = collection->size(); collection->resize(1); * This code will initialize and declare previous values to collection size. It will resize containers to hold elements. |
| **10** | * [test to verify clear erases the collection. Test to verify erase(begin,end) erases the collection. |
| **11** | AUTOMATION SUMMARY |
| **12** | A DevSecOps pipeline, which is a CI\CD pipeline with integrated security practices and tooling, adds practices and functions like scanning, threat intelligence, policy enforcement, static analysis, and compliance validation to the software development lifecycle (SDLC). Instead of tacking security on to the end of projects with point-in-time audits and penetration tests after code is deployed, DevSecOps bakes security in at every step of the process. This includes building, testing, and deploying software where security was often an afterthought. |
| **13** | It was reported that it costs 6 times more to fix a bug found during implementation than to fix one identified during design  The price to fix bugs during the testing phase can become 15 times more thab the cost of fixing those during design  Even though coding security best practices and secure coding standards are not overly complex, it can nevertheless be a challenge knowing when and how to use the right coding security best practice or coding standard. |
| **14** | * + - Security Layers and use of Defense in Depth.     - Having a data back up every other week should provide more security for occurring data loss.     - If something will happen to a physical device with all of the information and it won’t be accessible anymore we should be able to access the backups.     - There must be access management for every company.     - The reasons for that is giving admin permissions only for the installation of necessary for work applications or none at all. Some companies d not allow admin permissions for the employees and they require permission from management or IT department to install software. |

**References**

Michali. (2022, September 15). *What is a DevSecOps pipeline?* Check Point Software. Retrieved February 2, 2023, from https://www.checkpoint.com/cyber-hub/cloud-security/devsecops/what-is-a-devsecops-pipeline/#:~:text=A%20DevSecOps%20pipeline%2C%20which%20is,software%20development%20lifecycle%20(SDLC).

*Guide to coding security and secure coding standards*. Perforce. (n.d.). Retrieved February 2, 2023, from https://www.perforce.com/p/kw/secure-coding-standards-guide?utm\_leadsource=cpc-googleadwords&utm\_source=googleadwords&utm\_medium=cpc&utm\_campaign=KlocworkNA&utm\_adgroup=Secure-Coding-Standards&gclid=EAIaIQobChMIlOzV-\_z3\_AIVT1JyCh3IxgdaEAAYAiAAEgJ2lfD\_BwE