Dylan Coulter February 27, 2025

ID#1896062

C405

# CS 405 Project Two Script Template

YouTube Link: <https://youtu.be/0cHTsE2iIgw>

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello, and welcome everyone, my name is Dylan, and I will be presenting the Green Pace security policy to you today.  I will be going into detail about different topics such as Defense in depth, encryption policies, triple A policies, coding standards, unit testing, the importance of automation, risks of acting now vs acting later, useable tools, recommendations, and to finish a conclusion. |
| **2** | A crucial part of securing software starts with having a good understanding of defense in depth. Defense in depth focuses on providing many layers of security to a project throughout development. Using one layer of security may work fine to protect one section of a project but lacks with helping to protect other areas. This is why the use of multi layered security is important. It is important to consider how many layers of security is needed because more layers mean more cost and energy, so it is important to find a middle ground. It may not be efficient to build something with many layers to protect data that is not sensitive. |
| **3** | A threat matrix chart can be used to organize and display the coding vulnerabilities to help gain a better understanding of the severity of each threat, and which order they should be prioritized. With this a company looking to make security changes can easily identify the coding vulnerability that is most likely to happen and the most severe. Then they work their way down the list where red indicates an immediate response should be taken to green indication it is not as high of a priority and can be looked at later. |
| **4** | There are 10 coding standards that play an integral role in helping programmers  understand and identify different areas that should be considered when building software with DevSecOps in mind. Within these 10 coding standards coding risks have been paired with these standards and some coding risk fall into multiple standards categories. With the proper use of the coding standards vulnerabilities will be reduced. |
| **5** | On this slide these coding standards have been listed in priority order. The highest priority is at the top of the list and there are some ties meaning they are equally as important and should be further looked at when considering which to work on first. A company should consider which parts will be affected by the vulnerabilities and consider the motives or possible attackers when choosing which to fix first. Starting at the top of the priority list is most important. |
| **6** | Moving on from the policies and standards, we will turn our focus to the encryption policies that will be implemented. Encryption at flight is focused on the utilization of encryption strategies for any data that is going to be transmitted over a network from a particular location to another. This can include any public communication channels or requests and or responses between a server and a client. Data in flight can more insecure that data at rest because it is exposed to the internet which make it a possibility for it to be modified and or intercepted. All information that that is going to be moving internally or externally source and destination must follow the encryption in flight policy. This means we must use standards such as AES-256 and also relevant protocols like SSL/TLS to encrypt and protect sensitive information. We must apply these protocols to APIs, email, web applications among other services. Encryption at rest is focused on the application of encryption strategies for inactive and stored data. This includes information in hard drives, databases, and any other long term storage solutions. The use of encryption algorithms can be utilized to mitigate any unauthorized access to malicious actors. Any data that will be kept for an indeterminate or substantial amount of time needs to follow the encryption at rest policy. This means all locally collected files, backup information, or database contents must be encrypted utilizing the AES-256 encryption standard. This will help to prevent any unauthorized access in the event that the data at rest had been compromised. Encryption in use is focused on the utilization of different encryption strategies to protect data this actively being used. This pertains to data that is being processed, changes, or read. Data at this point is most vulnerable. To prevent malicious actor’s encryption strategies must be utilized to safeguard and promote resiliency. Encryption in use policy must be applied to all information that is going to accessed by any users or systems. Principle of least privilege must be enforced to help protect the vulnerable data. This will help mistakes not have such an impact or corrupt the data. The use of the correct authentication standards with the addition of robust encryption protocols to decrypt data while it is needed and then later to encrypt it is imperative to assist in protect the data. |
| **7** | Next, we will be reviewing the Triple-A framework. The three A’s in triple A stand for authentication, authorization, and accounting. Authentication is a concept that focuses on the validation of whether the user is who they say they are. Protect systems should only be accessed be verified applications or users and authentication ensures this. The use of Multi-Factor Authentication can be utilized to confirm any access request. Other best practices can be followed such as the use of strong passwords can reduce the risk of malicious actors having the ability to be authenticated like a trusted user. The authentication policy must be applied by taking the passwords that will be used to log into the system and securely hashing them. This should also extend to potentially vulnerable or protected resources helping to prevent any unauthorized access due to any overlooked weak spots. To also help prevent the risk of attacks and minimize the surface area, time windows should be enforced when inactivity is detected on user logins. Authorization sets the different levels of access the users have to the system. This helps to prevent access user from have access to the entire system they may be authenticated, but that does not mean they will have the highest level of access. The principle of least privilege should be enforced or that users should only be granted the least amount of access that is required. We can use the principle of Default Deny to better help protect actions and resources that may have not been clearly accounted for. The use of role-based permissions must be applied to all users that engage with sensitive resources and systems. The ability to add new users must be regulated due to the fact the pose a risk of unwarranted outcomes if mistreated. Accounting revolves around the logging and keeping track of user activity and or the processes within the system. Some examples of this are logging changes made to database items and keeping track of the users’ activity when a protect resources has been accessed. These examples and forms of tracking and logging can be gathered into reports to promote efficient incident tracking. Accounting must be applied to all interactions and transactions with any resources or sensitive data within the system. Unusual activity or activities will be easier to catch if we are prudent about the logs and metrics that we are tracking. With this any incidents that arise can be adequately accessed to determine at which point the failure occurred. |
| **8** | Unit test are an important part of creating a secure and well function program. With testing individual units of code in the early stages of development, unit test can help identify and correct bugs before they become bigger issues and more costly to resolve. Unit test help to create cleaner and more reliable code by enforcing better coding practices ensuring every part of the application functions as expected. The ability to find bugs in the early stages of development. |
| **9** | The first unit test verifies clear will erase the collection. Here we  added 5 entries to the collection and then cleared the collection if the collection was clear properly it would result in ASSERT\_TRUE. If the was still entries in the collection after being cleared it would result in ASSERT\_FALSE. During testing this test passed. |
| **10** | The next unit test verifies erasing the beginning and end erases the collection. Here we once again added 5 entries. Then we erased both the beginning and end. If the collection size resulted in zero it would be ASSERT\_TRUE. If there were still entries in the collected the result would be ASSERT\_FALSE.  During testing this test passed. |
| **11** | The third unit test verifies that five values can be added to the collection. Five entries were added to the collection and if the result was originalSize equating to five it would result in ASSERT\_TRUE. If originalSize equated to not being 5 it would result in ASSERT\_FALSE. During testing this test passed. |
| **12** | The last unit test verifies if max size is greater than or equal to 0, 1, 5, or 10. The test will pass and result in ASSERT\_TRUE if if max\_size is greater than or equal to 0, 1, 5, or 10. During testing this test passed. |
| **13** | Automation will be included in numerous places in the DevOps cycle. On the pre-production end automation is going to be incorporated into the design, verification, and testing phases. Then on the production end, automation is going to be incorporated in the response, and monitoring and detection phases.  Security will be improved by introducing automation to the DevOps cycle while also assisting in helping to enforce new policies throughout the development process. This some of the beginning steps in transforming the DevOps cycle into a DevSecOps cycle. There are tools we will be looking to incorportate into pre-production, and they are Selenium, Sonarlint, and SonarCloud. The addition of these tools will play significant role with helping to identify vulnerabilities and testing for bugs by running unit test. Turning our focus to the production side we will incorporate automated logging to assist with identifying any unusual activity. If unusual activity is recognized automation will be used to block the connections until they are able to be verified and confirmed to be safe. |
| **14** | On this slide we will be covering the DevOps pipeline which is presented in the illustration on the right side of this slide and how DevSecOps is implemented. DevSecOps is built on top of the DevOps pipeline by implementing security procedures into every phase of the DevOps pipeline. Benefits of DevSecOps include easier monitoring and logging, quicker development, lower costs, better testing options, and greater software security protection. It is important all developers consider using this pipeline to promote improved productivity and security of their software projects. |
| **15** | There are many tools that can be used. One of which is Cppcheck which can be seen on the right. Cppcheck is a tool use for static C/C++ code analysis.  Some Similar tools include SonarQube, ESLint, DeepScan, CodeScene, Clang, Cpplint, and Codacy. Static code analysis tools are a key part of helping developers create efficient and secure software. |
| **16** | On this slide we are going to cover the risk and benefits of either acting now or acting later. Benefits of acting now include mitigation of current risks, cost savings in the long run, developers will be better trained, the high upfront cost outweighs the overall cost if a breach were to occur. Now, there are also risk with acting now and they include high cost upfront, the needed resources may not be available, increased training time for developers. Now on the other side there are also benefits and risk of acting later and the benefits include more time to gather resources and money, lower upfront cost as old equipment should suffice, cost saving due to the ability to hire less experience developers. The risks of acting later include higher cost to deal with issue when they are found later, extra testing time, trust implications if a breach were to occur, high cost if a breach were to happen. |
| **17** | There are many recommendations to be made and it starts with implemented DevSecOps immediately. Next, we must review the current software and identify least to most secure so we gain and understanding where to start and what is most vulnerable. We should ensure that our software developers are trained on the new security standards that are being implemented and security training should be continuous. We must review, revise, and edited the security policy regularly. It is important we implement Defense-in-depth layers as soon as possible. CERT coding standards should always be referenced while also implementing automation into testing, encryption, and triple-A policies. It is important we assess current and future threats and vulnerabilities. |
| **18** | It is important we act now and take the proper measures to properly implement security processes and standards. Not being proactive can result in unknown vulnerabilities and risks. Any successful attack can have major implications on an organization anywhere from the customer or clients trust all the way up to millions of dollars in fines and loss. It is important we implement DevSecOps, Triple-A, automation, and encryption policies to reduce risk. Promoting immediate action is important as the initial cost may be high, but they far out way the higher cost of future issues. Following this as a guide should be a great starting point in starting to gain the highest levels of security. |
| **19** | Resources – NO AUDIO |
| **20** | Resources – NO AUDIO |
| **21** | Resources – Image links  NO AUDIO |

Resources

# Berga, K. (2024, September 18). *The Ultimate Guide to Unit Testing: Benefits, Challenges, and Best Practices* <https://www.testdevlab.com/blog/the-ultimate-guide-to-unit-testing>

# Berga, K. (2024, July 03). *What Is Automated Testing and How Does It Work? (with Example)* <https://www.testdevlab.com/blog/automated-testing>

Knowles, M. (2024, December 06). *What is DevSecOps? 5 Key Components*

# <https://hyperproof.io/resource/devsecops/>

Monahan, R. (2021, August 11). *What Is DevOps? Definition, Goals, Methodology, and Best Practices* [*https://www.spiceworks.com/tech/devops/articles/what-is-devops/*](https://www.spiceworks.com/tech/devops/articles/what-is-devops/)

Prajapati, K. (2025, February 11). *AAA in Cybersecurity: Strengthening Security with Authentication, Authorization, and Accounting* <https://vocal.media/education/aaa-in-cybersecurity-strengthening-security-with-authentication-authorization-and-accounting>

# Raza, M. (2023, January 31). *Encryption Explained: At Rest, In Transit & End-To-End Encryption* [*https://www.spiceworks.com/tech/devops/articles/what-is-devops/*](https://www.spiceworks.com/tech/devops/articles/what-is-devops/)