# Project Two: Security Policy Presentation

Anaid Rodriguez

12/08/2023

Link: https://youtu.be/verL0faGvkE

| **Slide Number** | **Narrative** |
| --- | --- |
| **1** | Hello, my name is Anaid Rodriguez and today I will be presenting the Green Pace security policy guide. |
| **2** | I'd like to begin by addressing why the new security policy is needed. Although our previous policy was helpful, it was not thorough enough. We need a policy that addresses as many weak areas as possible and clearly defines preventative measures for cyber attacks. As our world transitions to become more technology dependent, our databases grow and the amount of sensitive data we hold grows with it. |
| **3** | Here, the matrix displays a brief summary of the security risks. Likely to happen risks are risks that are the most common although their level of severity ranges from low to high. A priority risk is a risk that is severe regardless of how likely it is to happen. These risks should be avoided at all cost and immediately addressed if found. A low priority risk is a risk that should be addressed after all the high priority risks have been reviewed. These risks may or may not be likely but do not pose an immediate security threat. Lastly, the unlikely risks are risks that aren't very common to come across but their severity could range from low to high. |
| **4** | The 10 principles are as follows: Validate input data, heed compiler warnings, architect and design for security policies, keep it simple, default deny, adhere to the principle of least privilege, sanitize data sent to other systems, practice defense in depth, use effective quality assurance techniques, adopt a secure coding standard. The coding standards are also listed here and those are: data type, data value, string correctness, SQL injection, memory protection, assertions, and exceptions. |
| **5** | The presented coding standards are organized in order of most important to least. The three standards that are the highest in the priority list are SQL injections, memory protection, and exceptions. The reasoning behind this is because SQL injections present a very common threat in cyber security from external attacks while vulnerabilities such as memory leaks and mishandled exceptions are common threats for internal attacks. This is not to say the rest of the standards are not important, but they are easier risks to prevent and easier issues to fix than the top 3. |
| **6** | In this slide I'll summarize the policies for encryption in rest, encryption in flight and encryption in use. The first policy covers encryption in rest. Data that is not being currently used (in other words, data that is at rest) is encrypted. This protects files that are no currently active but can still pose a security threat if accessed. It is just as important to encrypt data at rest as it is to protect data in flight and in use. In encryption in flight, data that moves through the system is encrypted. This data is usually the most sensitive data and most susceptible to vulnerabilities as it’s being moved through different areas of the program (and can therefore affect the whole system). Lastly, in encryption in use, data that is currently being used is encrypted. This data is the most susceptible to brute force or DoS attacks. It is crucial to protect this data because it is the most commonly attacked. |
| **7** | In this slide I'll summarize the policies that support authentication, authorization, and accounting. The Triple A framework strategy is an easy to implement defense that consist of 3 sections. The first A is authentication. This is the process of confirming the user should have access to the program. This is the first line of defense in the Triple A tactic. The second A is authorization. Authorization is the process of assigning certain roles to specific users. This is the second line of defense because after the user has been granted access, they need clear defined roles within the system. Last, but not least, the third A stands for accounting. Accounting is the process of tracking the users movements within the system. This is the final line of defense in the Triple A tactic because it can detect unusual or suspicious activity from the user. |
| **8** | The first unit test is a positive test. This test uses both ASSERT and EXPECT to test the effect of arithmetic operations. It begins by asserting the collection size is 0, which tells us it is an empty vector. Then, we add a single entry and expect the new size to be 1. After that, an arithmetic operation adds 5 to the size of the vector. The test then asserts the new size is 6 (due to the addition). The final two lines perform another arithmetic operation, which is multiplication, and the test asserts the vector size was appropriately resized. |
| **9** | The second unit test is a negative test. This specific test is used to verify a length error exception is thrown. It begins by expecting a vector of size 0 then asserts a length error exception will be thrown if we attempt to increase the size to 1 plus the max size. The program successfully threw the exception and the last line is added to ensure the vector remained unaffected. |
| **10** | The third unit test is another negative test. This test also tests that an exception is thrown but in this case it asserts an out of range exception is thrown. The test begins by verifying the size of the empty vector, then it adds a single entry, next it verifies the vector is now of size one and the last line asserts the exception is thrown when we attempt to access the vector at size 5. |
| **11** | The last test we have here is a positive test. This test is simple and is used to verify that clear successfully erases the collection. The first step in the test is to verify the collection is empty, then it adds 10 entries. Next, we verify the vector was successfully resized and then clear it. The final step is to verify the collection is now fully empty. |
| **12** | This slide provides an image of the DevSecOps which explains the overall automation summary. |
| **13** | The image in the previous slide is of the DevSecOps which is a pipeline that integrates security practices and tools to comply with the software development lifecycle (SDLC). DevSecOps has proven to be an effective way to deliver secure and high-quality software. External tools are used in both the pre-production and production phases. In terms of the diagram, in pre-production, external tools are used in the sections “assess and plan” and “verify and test”. In the production phase, external tools are used in the “monitor and detect” section. Vulnerabilities in these sections are detected through automated testing, then the found vulnerabilities are addressed and worked on. Risks found during this testing can be those that entered through the pipeline or from an external source. |
| **14** | I believe there are little to no benefits in waiting to act. Acting now can help us gain a head start in security and gives us more time to fully understand and familiarize ourselves with the code we are developing. The current strategy is lacking initiative but does have the benefit of being less impulsive. The main risk of acting now rather than waiting is running of the risk of not having all the information needed to develop and test. However, I believe the risk is minimal and we should begin the implementation. |
| **15** | The first and most important recommendation is: perform constant testing. This is because the main gap in the security policy is the constant changing technology. As I mentioned in the beginning of the presentation, we are increasingly becoming more dependent on technology which means it’ll keep changing and innovations are not only expected but demanded. We must be prepared for new developments because with new developments come new, never before encountered threats. Additionally, do not leave any risks up to chance, always assume every single risk will be exploited and plan to defend it accordingly. |
| **16** | Cyber attacks are dangerous and can hurt a lot of people. It’s important that we take security seriously and help prevent damage to users and companies. Applying safety protocols and adopting secure coding standards can build the perfect wall of defense against malicious attacks. That’s all for this presentation, any questions you may have feel free to reach out to me. It was a pleasure to present the new security policy and thank you for your time. |