ISEC 400 Homework 2 Name: Megan Leonard

Answer the following questions based on your reading of the textbooks, any supplemental material, and the instructor’s presentation this week. If you use an external source (i.e. a web-page, the required textbook, or an additional book) to help you answer the questions then be sure to cite that source. Hint: you should probably always be citing a source.

## Questions

1. **[5 points]** How is a secure software development lifecycle different from the standard software development lifecycle? What specific activities are in each? Describe at least one secure lifecycle model.

A secure and standard software development lifecycle have the same 5 phases of requirements, design, coding, testing, and maintenance. The outside looks the same but looking closer at the steps you can see how they can differ. Requirements for both looks at the overview of the projects but the secure lifecycle has additional requirements that include security requirements, risk assessment, and requirements inspections. The design looks at how it is made as well as different security measures such as least privilege, defense in depth, and separation of privilege. The coding will follow with securing the weakest link and defense in depth as well as creating code to handle the common threats like phishing and malware. Testing will make sure that if it fails, it will do so securely and test out attacks against the software. The maintenance will look at fixing any found holes in the software that were located through the testing. One secure lifecycle model is the spiral model which will travel in a spiral pattern through each of the phases until it reaches the desired outcome. The spiral is also a way of showing the cumulative cost as the larger the spiral gets, the more expensive it becomes.

1. **[10 points]** The SQUARE methodology (see <http://www.sei.cmu.edu/reports/05tr009.pdf>) for security requirements elicitation from CMU lists 9 steps. Describe what happens in each step. Is there anything missing from this process that you would add?

1. Agree on definitions  
The first step looks at how different areas or things are defined between the stakeholders and the creators. This is to make sure that everyone has the same understanding of key words and phrases to not cause confusion moving forward.

2. Identify security goals  
The second step looks at the goals that are put into place by the stakeholders and the overall business goals. The goals are to be looked at with the existing policy to determine if the goals work or conflict. There are also use and misuse cases which take not of events that could happen and their effect.

3. Develop artifacts  
The third step starts on creating the system architecture diagrams and attack trees. This looks at the goals and policies from the prior steps and implements them within the design.

4. Perform risk assessment  
The fourth step looks at the probability of different threats and determines vulnerabilities, the impact, and how to minimize the chance and damage.

5. Select elicitation technique  
The fifth step will collect information on how the users interact with the software and the stakeholders.

6. Elicit security requirements  
The sixth step looks at what the systems are capable of and its measurable requirements of what it can do and handle.

7. Categorize requirements  
Step seven looks at the requirements that need to be met and puts them into categories using ones defined from the stakeholders.

8. Prioritize requirements  
Step eight looks at the requirements that have been determined and decide which ones can be implemented with the time, technology, finance, and ability that they have.

9. Requirements inspection

This final step looks for any issues within the determined requirements such as wording or feasibility. The inspection looks at it in a formal practice with those who have a technical background and then informal which is shown to those who may not understand the technical aspects as well.

I think that the 9 steps covers the process quite well and if I was to add anything it would be taking time to write out the limitations of the project prior to the requirements discussion. For example, if you know that the project will not work on the cloud, it is best to let the stakeholders and others know first as a good baseline of the starting requirements.

1. **[10 points]** The eight design principles for secure software discussed in class are highly related. How does each of the principles relate to the others? How do these principles relate to the SQUARE method from the previous question?

The least privilege principle starts with making sure that a person only has minimum necessary access which relates to the separation of privilege as the work is divided up into the different access levels. The defense in depth is layering the security as a means of defense so it one fails, the other is still there which relates to the fail securely. Fail securely makes sure that when a system fails, it still protects the sensitive information. The securing the weakest link looks at the weak points and determine how to fortify them and make them more secure which helps the system fail securely and involves the defense in depth. Separation of privilege makes sure that the access is given to users in their distinct area. It helps split up the work areas and makes sure no one person has access to everything. Simplicity looks at the security and makes sure it is not too complex as the more complex it gets, the more vulnerabilities it creates which relates to the no security by obscurity and securing the weakest link. Usability looks at accessibility and makes sure that those with access to areas can get to those areas. This connects to the no security by obscurity as the security needs to not cause problems with accessibility and separation of privilege as those who can access areas are limited and not everyone has the same access. No security by obscurity looks at how the implementation and design need to be accessible to those who work on it and hidden parts can cause issues when trying to prevent an attack. This connects to the simplicity as it needs to be not obscure and complex when unnecessary. The principles come back to the square method as they can be looked at as rules that the requirements should adhere to. The principles are easy to follow steps when determining certain requirements such as a system cannot be obscure or too complex. It gives a hand with the requirement prioritization by comparing them to its principles.

1. **[5 points]** In approximately 300 to 400 of prose (i.e. sentences, not bullet lists) using APA style citations if needed, summarize and interact with the content that was covered this week in class. In your summary, you should highlight the major topics, theories, practices, and knowledge that were covered. Your summary should also interact with the material through personal observations, reflections, and applications to the field of study. In particular, highlight what surprised, enlightened, or otherwise engaged you. Make sure to include at least one thing that you’re still confused about. In other words, you should think and write critically not just about what was presented but also what you have learned through the session. Ask at least one question that your instructor can answer in the returned assignment or class discussion.

This week we at the design principles and the security that can be done during the design lifecycle. I always find it interesting to see how past topics will always come back to connect to the new topics we are learning. The software lifecycle is one that comes back a lot during these classes and seeing it compared to a more secure version was fun. The requirements was new and it is interesting to hear about the shareholders and the say they have within business. We mostly hear about the programmer side so hearing more about the business is always interesting. My question for the week is what is the weirdest thing you have hold of a shareholder requesting of a tech company or person?

Citation:

Reddy, G. C. (2021, June 29). *Spiral model*. Software Testing. Retrieved February 4, 2023, from https://www.gcreddy.com/2021/06/spiral-model.html

Sullivan, B., Liu, V., & Howard, M. (2012). Web Application Security: A Beginner's Guide. McGraw-Hill. ISBN: 9780071776165

Whittaker, T., &amp; Whitney, T. (2022). Week02 Power Point. Retrieved February 4, 2023, from <https://franklin.instructure.com/courses/11070/files/1516715?module_item_id=623549>