Upon reflection of this course, there were a lot of interesting things that we covered that have changed how I will look at things going forward in development. When looking at how important security is I can understand the importance of working on security throughout the development of the product. If we leave security to the end, we can risk creating an overall inferior security system versus if we planned it out and worked on implementing it from the beginning. However, we also need to create a plan for that security and that is where having a secure coding standard comes into play. While working on software I will plan on how I wish to implement some types of security and operate off that plan. By getting feedback on the plan from others working on the project I will have the ability to improve the security of our application or software.

The other part that we covered was the assessment of risks and the cost-benefit of mitigation. When we looked at some of the examples of hacks that caused companies to lose a lot of money and consumer trust it shaped the way I would look at the risk assessment. I would prefer to still work on implementing any fixes even if the risk is relatively low, however, we can use that risk assessment to create an order and plan of what we should be working on and when. It will be important to analyze the cost of mitigating vulnerabilities that have a low potential, but I would still wish to push for them to be worked on.

The other party that we discussed in class was zero trust and going forward I will implement a zero-trust type of development. I would want the users to have to have secondary authentication methods like 2FA or security keys as well as not just default into giving someone in our network access to everything. Going forward I will approach anyone trying to get access as if they aren’t supposed to have access and work on ways to implement checks that verify, they should have access and limit the access that they must what is necessary.

Lastly, I would like to discuss the implementation of security policies and recommendations. From this class, I can see the importance of implementing changes in a timely manner. Just recently Twitter's linking abilities were down because a vulnerability that was disclosed was never worked on. Because of this, the person who discovered this vulnerability decided to make it public causing Twitter to finally work on implementing a change. We shouldn’t need to wait until someone makes a vulnerability public for us to fix it. We should look at reports from users and work on implementing fixes immediately rather than waiting for something to be public knowledge. This allows bad actors to use that public knowledge of the exploit to craft an attack due to our inability to properly react to user reports on vulnerabilities.