By having a secure coding standard, it will give the developers/designers of a program the minimum basic guidelines of what practices need to be implemented during the development of a program. In some cases, these basic practices should always be followed and shouldn’t have to be specifically called out, but in most cases, it helps ensure that all individuals are at the same place when it comes to the development process. By having a secure coding standard, basic security practices could be outlined in the policy so that they can be added into the program during development. By adding basic security safety features during the build process helps the testing process and health check part of the build process. If security in the program isn’t though of until the last phases of the development process, then the project could have to be completely or near completely redesigned. Sometimes adding some security features or security tests could impact how the program functions. So, adding these early on will help reduce the amount of time spent performing this task and prevent the chance of having to complete this during the final stages of the build.

If proper evaluations and risk assessments are not completed, then there is no way that a program that has security implemented in it could be programed properly. Yes, the program could have security implementations in place, but it would not necessarily have the correct ones or all the features that is needed. At the same time if proper assessments are not completed there is no way to properly determine the benefits that come from having a properly secured program. This would mean that there is no was to check the cost/benefits. Either way in most times if security is not thought of during the development process and is not implemented until the end. This could cause added time to be needed to complete the project. This being the case the program would have the security features, but the developer’s reputation could drop. This could mean that jobs they get would be reduced. This means that the cost of them waiting until the end for security was added development time and a mark on their reputations. The benefit that they would have gotten was reduction in the early stages of development due to not having to consider security concerns. Though this is a very crude and simple example the concepts still apply and are true.

Having a zero-trust policy currently makes since. Verifying all users and programs that access a system should be a requirement. There is so much data out on the internet and on servers that having proper security in place is necessary. One wrong person or one malicious download could currently lead to thousands if not millions of people’s data to be released. This shows that having a zero-trust policy for access to these programs and system should be the minimum necessity. Being able to ensure that only the proper people are accessing sensitive data will help ensure the security of it and minimize the risk of it being used in a malicious manner.

Personally, after performing the work on the security policy project, I realized how difficult it is to develop one of these policies. It also allowed me to see some of the reason why these standards and policies are created. If a security policy isn’t implanted, then there is no base line for a system when it comes to security. There are a lot of different parts and systems that work together for a facility to work properly. If the security for these systems is not designed properly then these systems could malfunction, and damages could occur. Implementing and following the security recommendations in a security policy allows for an increase in security awareness. This will also allow for people to have a better understanding of the possible threats out there that could affect their work.