**Defense in Depth (DiD)**

James Porter

Department of Computer Science, Southern New Hampshire University

CS 405 Secure Coding

Professor Wasim Alim

September 4, 2022

**How Deep is too Deep?**

When it comes to writing code, it is important to consider defense techniques from multiple different angles. It is also important to make sure that as many potential vulnerabilities are protected as possible. However, there is a line that can be crossed when too many resources or time are put into securing code. Even though we learned in week one that DiD means that we are defending from attacks through many techniques that protect from similar attacks, we can get to a point where some defense techniques are never used. For example, if we were checking if a number was less than zero and first checked if the number was less than zero and subsequently checked if the number was less than -6, we would never get to the second check.

**Considerations**

When considering defense techniques, there are some limitations that could change which tactics we use. First it is important to consider time constraints. Some tactics are very lengthy and require that a lot of man hours be put into making sure that the code works properly. If not, the code might not work, or it could not catch all the vulnerabilities. It is also important to consider best practices. It is possible that there is a security technique that is tried and true and will work with very little effort. This also means that there is a possibility that there is a technique that should NOT be used. Another consideration that should be taken into account is that of size constraints. Some safety techniques may take up more space than others and lead to other security concerns.

**DiD Unique Aspects**

There are many different unique aspects to be considered with defense in depth that will change according to the project that is being worked on. If you are working on a banking system for example, you would want to consider security techniques that are focused on number inputs and details that might change if numbers were incorrectly dealt with. On the other hand, if you were working on a system that was meant deal with library interactions, numbers would need to be dealt with differently than if you were building a bank system. This is true no matter what type of system you were building and for whatever purpose.