CS-405  
Professor Andujar  
Michael Turco  
22 April 2025

Module Eight Journal

For this journal, we will be discussing the ‘Zero Trust’ policy, recommendations for security policies, and not leaving security to the end. To start, zero trust is a fairly new way to approach handling security, which involves treating all users and requests as threats and verifying their identities before granting them access to anything. Conventionally, users of the same network were inherently trusted, and were able to send requests to internal systems without a problem, while a firewall kept external users from accessing those same systems. However, nowadays, many users work from home or need to request system data outside of the local network, and the security now comes down to the system verifying that the request indeed comes from the person who they say they are. This change means that instead of securing a firewall and allowing the users inside, you are now trying to verify each user and request, regardless of where it originates from. This concept change removes a lot of the assumed trust from internal users or requests, and switches to verifying anything and everything, every step of the way.  
  
 The statement “Don’t leave security to the end” refers to how adding security features towards the end of development can lead to implementation problems and lead to a worse off security system compared to if you built it alongside the code and systems. Sometimes, some security features cannot be implemented with the current system, and some features would have to be reworked entirely to be made secure. Compare this to implementing security alongside the system functionalities, which would allow for a quick change in directions when something doesn’t line up. Think about it like trying to build a basement to a house when the house is already finished, versus making a basement when it is just soil. When it is just soil, you can easily see everything, plan ahead for the shape, not working in the dark, etc… When you are digging under a house, you are risking it collapsing on top of you, you can’t get big machines in there, and the whole process is a lot slower and needs more measurements. Implementing security in an already established system is a similar challenge, compared to doing it while the system is still being built.

Some steps you can take to prevent threats would be to keep the various dependencies your system uses up to date, perform static and dynamic testing, and follow a set of security best practices. Some best practices include validating all input, using the principle of least privilege, authenticating users and requests properly, and even encrypting important data to prevent from data leaks. Defense in Depth (DiD) is a cybersecurity strategy which involves implementing multiple layers of defense to protect a system from external and internal threats. DiD is effective because it forces attackers to overcome multiple overlapping layers of security, and typically if one layer fails, another redundant layer is there to prevent any breaches. Adding more layers of security will take time and money, and can end up making your system more complicated and harder to maintain in the future, depending on how developed the current system is. Each layer of security requires time and money to make and maintain, and may require additional security measures or checks for each operation to go through.