Throughout the CS-405 course, I’ve learned many valuable details on security during and applications life cycle. We’ve learned the importance of encryption and the different types, the Triple-A Framework and how to verify a user, the importance of unit tests (both positive and negative) and much more. I’ve known that security is vital for a successful application, but I found it very interesting how many different security concepts and how essential it is to continue security practices on all aspects of the life cycle, and how it should be implemented iteratively. I’ve also learned to not over-analyze or over-evaluate security for a given application and to only use what’s appropriate for the given situation as the return-on-investment value may not be worth the implementation. Each application is unique and so should the security principles behind it to keep it protected.

Zero trust is a security practices that intends to move away from perimeter-centric approaches and shift towards continuous verification of trust across every device, user, and application (Kueh, 2020). The pillars of a zero trust security policy include device, user, transport/session, application and data trust. These separate pillars are separated out so that if a user gains access to one pillar, they will need to re-authenticate and verify to gain access to separate pillars. Moving forward in my career as a software engineer, I will continue to take security as a focus when developing applications and will seek out companies’ security policies. At my previous employer as I reflect, I don’t remember seeing or hearing about concrete security principles or coding standards, so I think that many developers had some frustration with this after deploying code into production.

**Resources**

Kueh, InfoSec Insider Tony, and Tony Kueh. “A Practical Guide to Zero-Trust Security.” *Threatpost English Global Threatpostcom*, 15 Jan. 2020, threatpost.com/practical-guide-zero-trust-security/151912/.