This course has really opened my eyes into how important security is, especially in this the modern world that we live in. With big data hacks published monthly, and the plethora of data that’s available for each person on the planet, the risks are high. The teachings in this course really have made me re-evaluate my job direction!

A secure coding standard is something I think that every company that writes software, both big and small should have. The standard documents what pitfalls a developer may face and outlines a solution to those pitfalls. Best part is it brings security into the design and build phase of the project and relegates it as a first-class citizen, not bolted on in the afterthought. Building with this security minded focus inheritantly makes the application safer, as the ‘common’ security issues are already taken care of if implemented in your standard. Tied with good unit/integration testing, and a security minded focus during the test period ensures that the application is much harder to compromise. Especially in the world of software sales, having a secure coding standard makes your product more appealing to potential buyers, as you telegraph that your intention is to secure their data and not leave it up to the winds of chance.

Risk and risk factors are hard to sometimes see. This is because your inheritantly biased to thing the application you’re working on is ‘good’ and ‘secure’. It very well could be, but it’s hard to separate from this bias. This course has taught me that assessing risk is best thought in the mind of the actual attacker; what vulnerabilities and deficiencies are going to allow me to compromise the core application? This line of questioning helps the developer to break down those biases and start to identify weaknesses. The flip side to this is the cost benefit of making those fixes. It has been shown through countless data hacks that the cost of not mitigating these issues is huge. Especially when private user data is in play. The benefit to making changes to make your application safer, far outweigh the cost of a hack, the loss of the data, and loss of public respect for the company.

Zero Trust is another security concept that I do not think most companies consider. Zero Trust is a concept in which a company validates all forms of traffic, even if that traffic originates in its own network, and they believe the traffic is from a trusted source. While on the face of it, verifying each file/api/query transmission could be seen as tedious, it allows a user to validate the original caller and ensure a hacker is not masquerading as someone else in order to gain access to the system or its data. Zero Trust is especially important when dealing with a large network, as the number of manual reviewers that can search for this data does not grow with company size.

Finally, I think the implementation and recommendation of good security policies is a must in today’s world, especially around a good secure coding standard as mentioned in the first paragraph. A more security conscience company is going to take steps to limit exposure to their data and their customers data via extension. They also become more attractive to do business with from a consumer standpoint.