**8-2 Journal: Portfolio Reflection**

**Ryan DeBraal**

**CS-405-X6389 Secure Coding 21EW6**

Throughout this course there has been the progression of this idea that secure coding starts even before a single line of code has been written and the necessary assumption that anyone using your application is doing so with bad intentions. This is often a very discouraging idea for new developers who are eager to provide the highest degree of freedom and robust feature sets to their end users right out of the gate.

Unfortunately, the reality is a different story. As has been covered, security breaches are common, but their commonality makes them no less disastrous to the corporation and the victims when they occur. A single incident can hurt a corporation’s financial prospects, reputation, and future growth potential for years in advance; and the same incident can create chaos in a victim’s life.

The act of purchasing something online or in store with anything other than antiquated and eye roll inducing physical money comes with the accepted risk (as low as it seems) that our entire financial stability might be put in jeopardy if we just happen to be the unluckiest amongst many.

Customers are often provided with a year or two of “identity theft protection” following one of these egregious headline making breaches. It’s the “cost of doing business” for a giant corporation.

A person’s most private information is compromised, sold on the dark web, stored in tens of thousands of distributed databases, and utilized years later to the detriment of the victim; and we’ve become so accustomed to this, that we as consumers often don’t even bat an eye when our friend’s neighbor’s brother’s life is “ruined” by what is analogous to a mind bobbling game of Russian roulette that’s we’re *all playing, all the time*. A game where there might be ten, twenty or even a hundred million empty chambers in the proverbial gun but there is always that one bullet that will shatter a person’s dreams of financial stability.

This is why secure coding is so important and the reason why zero trust is not only the accepted norm, but the only acceptable practice when it comes to building applications that handle private, confidential, and restricted data.

A developer’s job is to deliver software that does what the client wants, but a developer’s *duty* is to build software with an unwavering focus on keeping people’s information safe.

