Phillip Thoendel – Module 12 - Compliance

This week I will be discussing two case studies from our textbook “The DevOps Handbook”. One is called “Providing Compliance in Regulated Environments”, the other is called “Relying on Production Telemetry for ATM Systems”. Both case studies come with some important lessons learned, I will discuss that below.

**Providing Compliance in Regulated Environments**

This case study focuses on being able to demonstrate regulatory compliance in a dynamic environment. Amazon Web Services hosts data for some of the largest companies in the world including Phillips, Pacific Life, General Electric and more. These companies are bound by regulations from an information security perspective.

It is the responsibility of Amazon Web Services to demonstrate their regulatory compliance for the sake of their customers. Typically, this is done with documentation like screen-caps of servers running and CSV files filled with config settings and logs. But due to the dynamic nature of Amazons Web Services DevOps patterns it was difficult to properly audit regulatory compliance. The dynamic nature of auto-scaling can make servers appear and disappear, thus making it nearly impossible to use the standard auditing methods.

The solution that Amazon Web Services came up with was to design their engineering requirements around the actual regulations, as a result, they we able to use telemetry systems to send like Kibana or Splunk and auditors were able to retrieve the necessary data at their leisure.

The takeaway here is that they were able to solve their problem by designing from the perspective of the regulations they needed to demonstrate compliance with.

**Relying on Production Telemetry for ATM Systems**

This case study has to do with fraud that was perpetrated by a developer and the manner in which the fraud was discoverd. It is worth mentioning that that, unlike the last case study, the names of the organization and persons involved are not disclosed. The study does state that it was a “major financial organization”.

In this study a developer wrote a back door in some ATM software that allowed them to put the ATM into maintenance mode and dispense cash without making a proper transaction. Somehow this back door evaded discovery during peer reviews. In fact, it was not even discovered until telemetry determined that ATMs were being put into maintenance mode at unscheduled times.

The takeaway here is that developers can build things that evade peer review. And these things can at times have malicious intentions. However, the development of robust telemetry tools can help to discover things that might be missed in a peer review.

**Conclusion**

While it is true that DevOps is helping to bridge the gap between traditional development and operations teams, we are still somewhat early on in this integration. This trend is likely to continue for the next several years. DevOps may help to solve some organizational problems but there are still relics of the “old way” that will serve as a hurdle for organizations attempting to integrate DevOps. This is especially true if an organization is required to demonstrate any kind of regulatory compliance.