Derek Livermont

Module 12

CSD 380

Regulatory bodies tend to be behind the times. This is especially true in the tech sphere where methods and standards tend to constantly be shifting with new innovations. The two case studies from the book, “Proving Compliance in Regulated Environments” and “Relying on Production Telemetry for ATM Systems”, were chosen to demonstrate how the changing technology surrounding applications requires new compliance and audit methods to effectively demonstrate that these organizations are following the law and protecting customers.

In the first case study, Bill Shinn from Amazon Web Services is tasked with managing his customers’ relationships with regulatory bodies. As someone who worked in the cloud space, he noticed that government auditors were using dated methods to collect evidence amongst thousands of auto-scaled cloud instances. Log samples, screenshots, and other data dumps were ineffective and overwhelming to parse to prove compliance.

The curious thing about this case is that, due to technological improvements, we live in a time where there is plenty of evidence that exists to prove that organizations are complying with laws. The problem was putting it in a place where auditors could get to it. Shinn started working with engineering teams to build telemetry and reporting into application production. Then, send that data directly into a collection source like Splunk or Kibana. Create access for auditors so that they can find the information that proves a certain guideline is being followed.

The case study also introduced the *DevOps Audit Defense Toolkit*, a comprehensive guide to preparing for the audit process as a DevOps organization. It does so by narrating the compliance and audit process of the fictitious Parts Unlimited. It even provides objections that auditors might theoretically raise and how to respond to them appropriately. It is generalized but can be adapted to common regulations such as HIPAA, SOX, and EU Model Contracts. The goal of this book as well as Shinn’s position is to show that DevOps can comply with regulations effectively.

I found the second case study, though shorter, much more compelling. It’s an interesting case as to why code reviews can be less effective than monitoring when it comes detecting vulnerabilities. This seems counterintuitive because code is the “source” of all actions of an application. However, as the study demonstrates, a developer was able to plant a back door into ATM code that allowed them to place ATMs in maintenance mode and withdraw money. This was missed in code reviews but they were quickly caught when unscheduled maintenance was reported by ATMs. The lesson learned is pretty on the nose. Sometimes, the regulatory measures put in place fail to catch bad actors. Newer methods like application telemetry can be more effective in identifying vulnerabilities. Visibility takes many forms and there is more than one way to secure an application.

Both of these cases demonstrate the changing landscape of application regulation. Methods that worked in earlier days of software development fall short of newer methods in the age of DevOps. It’s important that organizations that deal with these kinds of regulations understand how to advocate for themselves when working with auditors. In most cases, DevOps practices are more effective at securing applications than code reviews and separation of Dev and Ops duties.