Brett Fuller

CSD-380 - Assignment 12.2

7/25/25

**Proving Compliance in Regulated Environments**

In the case study on Proving Compliance in Regulated Environments Bill Shinn the principal security solutions architect for Amazon Web Services has the responsibility of showing customers they can meet compliance needs when embracing the cloud native services of AWS. While Shinn has worked with an astounding number of customers that publicly refer to their utilization of AWS public cloud he notes that a challenge many customers face is that auditors are not trained to validate compliance in ways that cleanly fit into DevOps models. Shinn provides an example of a traditional audit asking for a sampling of some portion of their servers along with details on asset management , and a host of other details that shows who has access to and what is happening on those servers. However, Shinn notes that this is no longer really viable when businesses are adopting models where servers are quickly being spun up and down in favor of new deployments. Scaling up and down. How do you provide a sampling of things that no longer exist? He notes that the traditional methods just do not fit the new models.

To address this Shinn and his team worked with auditors to devise a new strategy. Their design was to establish a single control in each sprint to provide necessary evidence in an audit. Providing on demand information on the systems while they are in production. In order to provide this data they leveraged their telemetry systems, so that by providing self service access to auditors. With these ideas in mind they found that by leveraging modern tools they could actually provide more efficient access to data for the auditors while requiring less effort from the staff during audits. Still this is not without challenges as there are a wide variety of compliance needs and understanding every aspect that may be required takes a lot of effort. Unfortunately the processes for identifying what is needed and how to gather it need to happen between multiple internal and external teams. From there gaining the data can happen in the CI/CD pipeline or from monitoring tools or other elements in the stack. In order to help with this process they developed the DevOpsAudit Defence Toolkit which shows how this process could be accomplished with a fictional example of a company meeting its compliance needs. Through this toolkit it shows how this fake company deployed the controls necessary to provide this information for a variety of common regulatory compliance standards.

**Relying on Production Telemetry for ATM Systems**

In Relying on Production Telemetry for ATM Systems we hear about a woman who leads a DevOps initiative for a large bank. This woman identifies that the organization is too reliant on Code Reviews for fraud detection, and proposes utilizing monitoring and automated testing alongside code reviews to minimize risk. She then talks about a time when a developer had put a backdoor into code for ATM deployments which allowed them to put the ATMs into maintenance mode so they could steal money from the ATMs. They caught the fraud not from a code review as she notes that these kinds of issues are extremely difficult to catch in a code review. Instead they found the issue by identifying that the ATMs were going into maintenance mode outside of scheduled hours. The use of monitoring and logging data helped them find the fraud even before the counts identified that the money was missing. They note that this issue occurred even with separation of duties and a change control process. While these are solid principles, layering on additional layers of security such as telemetry data can be crucial in identifying issues.

**References**

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook, 2nd Edition: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. IT Revolution.