Jonah Aney 12/21/25 CSD380 Module 12.2 Assingment: Compliance

The case studies *Proving Compliance in Regulated Environments* and *Relying on Production Telemetry for ATM Systems* show that traditional compliance and audit practices have trouble keeping up with modern DevOps environments. Both examples reveal that older compliance methods do not work well in automated, cloud-based systems. The authors point out that continuous monitoring and telemetry are necessary to prove controls work and to detect fraud.

In *Proving Compliance in Regulated Environments*, Bill Shinn, a principal security solutions architect at Amazon Web Services, explains the challenges auditors face with DevOps-driven infrastructures. Traditional audit methods were made for static, physical environments and depend on server sampling, screenshots, and manual data collection. These methods do not work well when infrastructure is defined as code and servers are constantly created or removed through auto-scaling. Deployment pipelines also make audits harder by combining development and deployment tasks.

In the case study, Shinn emphasizes the need to rethink how audit evidence is produced. Instead of relying on static documentation, AWS teams should collaborate with auditors during the control design process. By taking an iterative approach and focusing on one control per sprint, teams can identify the required audit evidence early and ensure it is available when systems are deployed. This collaboration helps bridge the gap between traditional audit expectations and modern development practices.

Centralized telemetry platforms such as Splunk and Kibana play a critical role in this process. By sending relevant logs and configuration data into centralized systems, auditors can self-service the evidence they need by querying specific time ranges. This provides immediate visibility into whether controls are functioning as intended. This significantly reduces the need for manual sampling. Modern logging, chat tools, and deployment pipelines also increase communication/transparency in production environments. All of this in turn, makes compliance easier to demonstrate.

A key lesson from this case study is the importance of translating regulatory requirements into technical controls. Shinn uses HIPAA as an example and explains that organizations must carefully analyze regulatory language to identify required technical safeguards and audit controls. Once these requirements are understood, compliance, security, and DevOps teams can determine how to prevent, detect, and correct issues with the use of configuration management, monitoring tools, and logging frameworks. Embedding controls into version-controlled configurations and automated pipelines enables continuous compliance, rather than relying on a document driven process.

The second case study, *Relying on Production Telemetry for ATM Systems*, highlights the importance of detection controls along with real-time monitoring. Mary Smith, the leader of a DevOps initiative at a large financial institution, observed that auditors and regulators often placed too much emphasis on code reviews for fraud prevention. She describes an incident where a developer embedded a backdoor into ATM software, which enabled unauthorized cash withdrawals.

Despite the separation of duties and formal change approval processes, the fraud was not detected through code reviews. It was actually discovered through routine operations monitoring when a team member noticed ATMs entering maintenance mode at unexpected times. This production telemetry revealed abnormal behavior and allowed the organization to stop the fraud before it was identified through scheduled cash audits.

The main lesson from this case study is that preventative controls alone are insufficient. Individuals with sufficient access and intent can bypass development safeguards. Production telemetry provides a critical layer of defense by detecting anomalies and enabling rapid response when they occur.

To conclude, these case studies demonstrate that effective compliance in regulated environments relies on automation, continuous monitoring, and collaboration among auditors, security and DevOps teams. By embedding controls into pipelines and leveraging production telemetry, organizations can meet regulatory requirements while enhancing security, transparency, and operational resilience.

References:

Kim, G., Humble, J., Debois, P., & Willis, J. (2021). *The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations* (2nd ed.). IT Revolution Press.