Rachel Theis

Module 12 Assignment

CSD 380

7.27.25

**Case Study #1: Providing Compliance in Regulated Environments**

Summary:

This case study explores the challenges and innovations that are occurring related to auditing in the DevOps space. Bill Shinn, Principal Security Solutions Architect at AWS, seeks to ensure all organizations, even at an enterprise level, comply with all rules and regulations, even when auditing processes seem like a herculean task. Shinn notes that typical auditing processes are rooted in physical technologies and more sequential or slower-paced delivery processes. When auditors face products or materials with code-based infrastructure, requesting access to the correct materials for everything that needs to be audited can be a near-impossible task, especially with automatically scaling code and CI/CD principles.

Shinn reimagined a new auditing process that aligns better with DevOps principles and enables regular and thorough auditing to occur, even for large enterprises. Instead of auditors obtaining data through requests, receiving data in the form of screenshots and CSV files, auditors work with teams in the control design process. A single control is assigned to each sprint to collect what is needed for audit evidence. Then, this control data is sent to telemetry systems like Splunk or Kibana, enabling auditors to access the information they need without having to manually request it. Shinn’s innovative auditing process for projects using a code-based infrastructure bridges gaps between existing auditing processes and the scale and pace of DevOps projects at an enterprise level. Auditors are able to access meaningful information quickly, leading to safer applications and compliant organizations.

Lessons Learned:

Lesson 1: Integrate the auditing process into the development lifecycle when possible

Lesson 2: Leverage telemetry of audit control data rather than manual requests for data

Lesson 3: Auditors should work with DevOps teams to ensure that auditing is properly integrated into sprints and that useful data is being gathered.

Lesson 4: Every organization (from small businesses to enterprises) can properly comply with regulatory and compliance laws and rules by implementing telemetry whenever possible and working with auditors to iterate auditing into a project’s lifecycle.

Lesson 5: DevOps, especially CI/CD principles, changes the flow of a project and may require changing secondary processes like auditing in order to effectively execute these processes.

**Case Study #2: Relying on Production Telemetry for ATM Systems**

Summary:

This case study recounts an example of ATM theft that had been orchestrated by a developer at a major banking institution. An employee who had access to the ATM code planted a backdoor allowing them to put ATMs into maintenance mode at certain times, then steal cash out of the machines. The banking institution was able to catch this fraud quickly because an operations review noticed the unusual and excessive timing of scheduled maintenance on these ATMs. This catch occurred even before the cash was audited and reconciled on the ATM side. This example shows the importance of automatically collecting data and developing systems to identify data anomalies. Furthermore, this case study warns against putting too much trust in manual code review, as code review can miss code vulnerabilities or intentional misuse, as was seen here.

Lessons Learned:

Lesson 1: Manual code review is prone to error and can’t catch all errors or malicious behaviors.

Lesson 2: Telemetry should be used whenever possible, and can find anomalies that human review could miss.

Lesson 3: Separation of Duties can help detect abnormal behaviors and provide some safeguards against unintentional errors or intentional fraud.

Lesson 4: Organizations should establish automatic data monitoring that can alert teams when anomalies occur.

Lesson 5: Organizations should hold regular review meetings to discuss data anomalies or other concerns.