**Assignment 12: Compliance**

Louis Capps

Bellevue University

CSD380 DevOps

Professor Darren Osier

27 July 2024

**Compliance**

 By not continuously improving how code is audited and reviewed, issues can occur with regulations and rules, as well as potential damages and security breaches. This paper will look at two different case studies. The first examines how outdated processes and procedures are unable to effectively examine code for compliance breaches, and the second looks at how an overreliance on code reviews leads to hackable or malicious code being deployed. Additionally, the paper will discuss the lessons learned from the case studies and how those lessons lead to better outcomes for auditing and fraud detection.

In the case study “Proving Compliance in Regulated Environments (2015)”, the author's main point is that auditors aren’t well trained for DevOps environments. They rely on outdated methods that work on physical environments to audit and check code for compliance, leading to a poor system for software to ensure that compliance is being met. Having alternative methods that work better is crucial to ensuring compliance with rules and regulations is being met. The lesson learned is that having not only a well-developed system to track what is going on in production but also a way to retrieve information from that system is critical to the auditing process and leads to a much better outcome of code compliance being met.

In the case study “Relying on Production Telemetry for ATM Systems”, the author's main point is that an overreliance on code reviews for fraud detection leads to potential security risks. Having only code reviews could lead to a potential breach that isn’t caught until major damage has already been done. In an ideal setting, production monitoring controls should be utilized in addition to the code review procedures already in place to ensure that those security risks are caught.

An example of how those monitoring controls, in addition to code reviews, can enhance security over code being deployed is presented in the case study. The author talks about how a developer purposefully introduced a back door into their system, allowing him unauthorized access when he wanted. This back door was missed during code reviews; however, monitor tools were able to catch him when he utilized that unauthorized access. The lesson learned is that with monitoring tools in place, the backdoor was discovered as it was occurring, and more damage would have occurred before the backdoor could be patched out.

In both of these case studies, reliance on old procedures and policies without implementing needed changes would or would have led to issues. In the first case study, it is a problem of compliance with laws and regulations. In the second case study, it is potential monetary loss from a lack of monitoring. By continuously developing new methods to better the systems we are working with, issues can be addressed and taken care of long before they become problems.

**Sources**

Kim, G., Humble, J., Debois, P., Willis, J., & Allspaw, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations* (First edition.). IT Revolution Press.