Chapter 23 Case Studies

This paper will summarize the case studies “Providing Compliance in Regulated Environments" and "Relying on Production Telemetry for ATM Systems" from Chapter 23 of *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations* by Gene Kim, Jez Humble, Patrick Debois, and John Willis. It will also discuss the lessons learned for each.

In the case study “Providing compliance in regulated environments”, principal security solution architect Bill Shinn created a DevOps audit defense toolkit. The purpose of the toolkit is to guide auditors and DevOps teams in making sure that their code is compliant with applicable regulations while also facilitating the auditing process. The reason for this is because Shinn noticed the traditional way of auditing was not cohesive with the fast-paced and dynamic nature of DevOps.

The lessons learned from this case study are that procedures are always changing and must continue to adapt, and we should not always rely on conventional practices. In the context of DevOps, the technological environment is always changing. Shinn saw that the existing auditing methods, which worked in more stable, physical environments, did not fit the fluid, scalable nature of DevOps. Technology is one of the most rapidly changing industries, so the ability to recognize changes being made as well as realizing that further improvements are missing, are huge factors. Instead of relying on the conventional practices that had been working in the past, Shinn implemented new changes to further develop the cohesion of DevOps and auditing teams. This change ensured that the auditing process could keep pace with the dynamic nature of DevOps.

In the case study “Relying on Production Telemetry for ATM Systems” a woman realized that the large US financial operation she worked for was placing too much faith in the ability of code reviews to detect fraud. This reliance may have contributed to the fact that a developer within the organization ended up planting a backdoor in the code to give them an ability to steal money from the ATM machines. The fraud was thankfully discovered quickly through an operations review.   
 The lessons learned from this case study are the code reviews are not sufficient enough to prevent threats, and multiple cyber security principles should be put into place. In this case study, the threat actor was a developer, so this automatically makes the practice of code reviewing simply not enough to prevent fraud. Multiple cyber security principles should be put into place, such as continuous monitoring, access controls, and automated alerts. These practices will help facilitate early detections in scenarios such as the one depicted in the case study. Thankfully, the team performed regular operation reviews and were able to catch the fraud relatively early. If they had implemented more security controls, they may have been able to catch it even quicker.

In conclusion, these case studies illustrate different important themes within the realm of DevOps. The first case study highlights the need for adaptability when it comes to auditing and team cohesion. It also shows us that traditional practices need to be revisited from time to time in order to make way for more productive methods. The second case study shows the importance of having several security practices in place, as well as the importance of not becoming too reliant on current practices. These two case studies highlight important trends that can lead to improved DevOps practices.

Works Cited

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations*. IT Revolution Press.