Austen Rhyce Erickson
CSD380: DevOps
Assignment 12.2: Case Studies
3/07/25

The case study called Providing Compliance in Regulated Environments talks about challenges with traditional audit methods such as using screenshots and sampling manually. As technology progresses, these methods are no longer suitable – especially in modern DevOps environments where infrastructure is managed as code and everything is dynamic. The solution has been to begin automating the collection of compliance evidence by integrating audit data into telemetry tools. This allows auditing to happen on demand in a self-service fashion. Some examples of tools that do this are Splunk and Kibana. To begin making this transition, it is recommended to collaborate with auditors early on in the design process and utilize deployment pipelines and logging frameworks to automatically generate the auditing evidence. There was an example of HIPAA's audit controls that translate regulation requirements into actual engineering implementations. Also discussed was the DevOps Audit Defense Toolkit which provides a framework for integrating compliance into deployment pipelines. This case study demonstrates that DevOps can meet compliance requirements and do so efficiently. Visibility can successfully be enhanced, risk can be reduced, and compliance can be proven in real time.

The case study entitled Relying on Production Telemetry for ATM Systems explores solutions to detecting fraud in ATM software. Like in the previous case study, traditional methods were ineffective. Code reviews and change approvals were no longer adequate. The noteworthy incident that demonstrates this was a case of a developer planning a backdoor in the software to allow unauthorized cash withdrawals. They were able to bypass code reviews and approval processes to get the code deployed to production. The fraud was eventually detected thanks to production telemetry. There were unexpected ATM maintenance mode activations that raised eyebrows. It turns out that regular operational monitoring was more effective than the static security controls. The takeaway here is that blindspots can occur when there is over-reliance on

code reviews and separation of duties. These blindspots can be taken advantage of by evil doers. Continuous monitoring and telemetry protects against this as it offers real-time visibility into system behavior. It's like a big eyeball watching and detecting anomalies as they occur.

A common theme occurring in both of these two case studies is that security methods need to evolve beyond the traditional status quo methods. Bad actors eventually become aware of loopholes and workarounds and develop ways to take advantage. Security must evolve and change to stay ahead of these bad actors. Ideally, it becomes so hard to hack systems that hackers don't even bother. This kind of dynamic adaptation required of modern security is found in good DevOps environments that have automated logging and telemetry to provide strong, real-time assurance of security and compliance with the best practice standards recommended by security defense experts. Another common theme is that collaboration between security, auditor, and DevOps teams is a must. Together, they can ensure regulatory requirements are met with the utmost efficiency. Lastly, the benefits of real-time production monitoring cannot be overstated. It is a critical measure for detecting fraud and operational issues as they occur.

Source:

Kim, Gene, et al. *The Devops Handbook How to Create World-Class Agility, Reliability, and Security in Technology Organizations.* It Revolution Pr, 2015.