Violet Gonzalez

Module 12

March 8th, 2025

The Bridge of Two Case Studies

In DevOps, there are many different aspects to consider. Additionally, these other concepts and practices can drastically alter an organization. Therefore, it is essential to understand what these practices and ideals mean and how they work in the bigger picture. Sometimes, an organization may experience a dilemma in its operations, and finding solutions to those concerns becomes the next big thing in DevOps. Two case studies display clear indications of corrupt or faulty security, rather than through their lack of data for audits or review. From these studies, we can understand more about how security is essential throughout the aftermath of development and how it can be utilized through tools and communication.

                In the case study, Proving Compliance in Regulated Environments, providing auditors with the information they need can prove difficult when non-traditional outlooks of development are taken. Amazon has once again come to light, but not for a bad reason; since they are constantly innovating, they run into the beginning issues of an idea first. Because auditors required specific information such as logs, screenshots, and other data, it was hard to provide data on how they deployed their software. To bridge the gap, the principal security solutions architect of Amazon Web Services, Bill Shinn, was able to combine development and operations through documentation to help better assess their data and provide it for auditors to review. This may seem like a rudimentary solution, but it takes a more complex correction. Understanding regulations from auditors can be complicated and, if anything, very nuanced; however, there are many ways to bridge that knowledge gap, and that is through proper documentation. Shinn realized the best way to accomplish the task of allowing the auditors to get the information they needed when they needed it could be provided through a telemetry system. A telemetry system enables an organization to collect data, thus providing them with the opportunity to monitor and control systems. Therefore, each portion of the development process was recorded using this system, including logs, chat rooms, and other things that could be considered. When an auditor needed to access the data, he had to log in to that telemetry system and find the data they wanted.

                However, getting to this point was not such a straightforward task. As mentioned, the original idea sounds very rudimentary, but getting there involves a bit more. This could circle towards the accomplishment of communication. It may seem a bit sarcastic, but it truly makes a difference in how much communication can change in the organization. Because Shinn Had this idea in mind, he needed to be able to execute it correctly. With that, he managed to have development teams work with auditors in the "control design process" (Kim, Humble, Debois, Willis, 2021) to ensure that their new way of collecting and documenting data would be sufficient for auditors. The principal security solutions architect's actions show how compliance and regulation can be combined and accomplished through documentation, thus improving risk assessment and mitigation. A big takeaway from this, besides the documentation portion, is that rules cannot be completed or met with just one clear-cut answer but instead "require alternative methods of presenting data" (Baugh, 2014) and Require some depth of understanding of the regulations on how to meet said requirements.

                Another case study, Relying on Production Telemetry for ATM Systems, carries the same embodiment or feeling of regulations or the idea of security. The difference between this case study and the prior one is the execution and how the security vulnerability was discovered. In our first case study, there wasn't much vulnerability in the library, but there was a lack of documentation or proper documentation for auditors to review. In this case study, we are presented with information about an organization with more documentation, which later benefits them. Revisiting the tool telemetry service, this specific banking property was able to locate a tough and discredited vulnerability that has been embedded within their software. There is a term that a lot of developers should be familiar with, which is the back door. Having a back door in your code or software means that a developer or somebody with access to that code has left an opening for them to re-enter-reenter. This is seen in the most recent Linux controversy, where a back door was found to allow a person with access to the back door to enter inside code and gather personal information and loads of other data collected through the software. However, I digress. The idea remains the same, where a back door is never used for a good purpose and almost always has a malicious intent.

Keeping this in mind, this organization followed audit regulations procedures quite meticulously; however, they weren't constrained to the ideals demonstrated by auditors. Therefore, when it was time for them to review code or talk about software, it wasn't just relying on that code review, nor was it a separation from the developers and the operations team but instead at integration. Because this organization not only had frequent discussions amongst them as a whole but utilized "effective production telemetry" (Kim, Humble, Debois, Willis, 2021), they quickly found this fraud attempt despite their black and white demeanor towards auditing compliance. Therefore, when there is a separation of duties, having a service such as telemetry allows for its high gap to be bridged and detect vulnerabilities and errors that may be missed by one or another team.

There is a commonality between these two case studies with telemetry; however, that shouldn't be the primary focus. Instead, focus on the ideals of security and regulation. Each organization had its flaws but overcame them with certain instances that they implemented. When conducting audits, it's essential to have proper documentation as it can lead to a paper trail or something to reference when something doesn't align. Additionally, following regulatory audits through proper compliance allows the company to withhold the integrity of its software better since it meets proper requirements. However, this is not to say that there is one way to align with the regulations that these auditors set. As seen with our second case study, it was through telemetry that they discovered a very discreet vulnerability that could have caused a great commotion within their organization. There's an overreliance on the black-and-white side of compliance with audits, which is not always the answer. A good approach would be an integration of both, which is more apparent in our first case study.

The solution to these case studies may not be simple; however, at the end of the day, it should utilize something that can display proper communication between developers and the operations team. Strict reliance on auditors can be a downfall for any company, yet not complying with auditors can also produce the same results. Therefore, finding a balance between understanding those regulations and implementing them within your organization can be key when considering better securing and complying with regulations.

References:

Baugh, A. D. (2023, March 14). *Compliance*. Medium. https://medium.com/@ace.d.baugh/compliance-170a3dafd1b5

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