Alisa Steensen 7/25/25

DevOps Module 12.2

Providing Compliance in Regulated Environments

When companies like banks or hospitals try to upgrade their tech, there are lots of strict rules they must follow, and that makes it tough to try out new things like DevOps or cloud services. This case study shows how that problem can actually be solved. Bill Shinn from AWS figured out a new way to handle compliance that works better with modern tools. Instead of letting audits and paperwork slowdown everyone, they built compliance checks right into the development process. So instead of waiting until the end to see if everything is okay, they automated those checks as part of the pipeline. Essentially, it’s proof that if you’re willing to change how you think about compliance, it doesn’t have to be difficult.

One of the biggest problems in this case study is the disconnect between how audits are traditionally done and how DevOps actually works. Naturally, old school audit methods are not a great fit. That is where Bill Shinn and his team come in. Instead of forcing DevOps to slow down for compliance, they decided to make compliance part of the process. There are a few lessons to be learned from this case study as follows.

Instead of waiting until the end of the development phase to verify compliance, the organization proactively involved auditors during the design phase of their DevOps process. This type of approach ensured that compliance controls were not only clearly defined but also integrated directly into the architecture from the start. This helped prevent miscommunication later in the process and allowed for a smoother process as controls were built with auditability in mind from the beginning.

Bill Shinn stated that to follow HIPPA rules, teams need to really understand what those rules say, not just guess. One they understand the rules, they should set up tech tools that match the rules. Then, they need to keep clear records that show those tools are working. For example, if the rule says to track who accesses patient info, the team should set up logging that shows that activity. And then by saving those logs in a way that is easy to find, it helps the team. This makes the audits a lot easier and shows that the system is safe.

Because many auditors are used to an older way of checking systems, that doesn't always work well anymore because of cloud systems. So, auditors need to know and use new tools and training to keep them up to date. Companies will have to help them learn how things work now. That means giving them access to live systems where they can look at real time data instead of waiting for someone to send it to them. It also means making sure there are clear logs and reports that go with the rules and requirements. This will ultimately save everyone time during audits.

Lastly, being open and clear about what is happening in your system is actually a great advantage. With today’s DevOps tools, almost everything is recorded automatically. For example, chat logs can show team discussions, version control can show exactly when code was changed, and deployment tools track what gets released and when. All the logs from these systems can be stored in one place, which makes it easier to find answers when something goes wrong. Additionally, it helps when auditors need to make sure that rules are being followed. The idea is making sure that all the data is collected properly and shown in a way that makes sense.

Relying on Production Telemetry for ATM Systems

This next case study discusses why watching what is happening in your systems is super important. In the case study, a developer put a backdoor into software that runs an ATM machine. This code let them flip ATMS into “maintenance mode” when they weren’t supposed to, and then they could take money out. The bad part is that no one noticed in the code review, and even though the bank had all the usual rules in place. This is especially important for places where fraud can happen. Mary Smith, who leads a DevOps team, explains that just having code reviews and approval steps isn't always enough to stop bad stuff. If someone has the right access and knows what they’re doing they can still sneak things in.

Because of good monitoring tools and regular team checks, someone was able to notice that the ATM in one particular area was acting strangely. It kept going into maintenance mode at night. That looked odd, so they were able to investigate and caught the problem early on before the usual money checking process began. There are some great points made in this case study and lessons learned, we will further discuss.

Even though traditional controls are in place, they can still be bypassed. These rules alone aren’t enough to stop someone who really wants to do some damage. If a person has the right access, or knows the system well, they might still be able to slip harmful code into production without anyone noticing at first. That is why it is important to have strong monitoring and regular checks in place. Check to ensure there is no strange behavior or if anything looks suspicious.

Mary Smith also made a great point that developers should be relying on production monitoring controls in addition to testing, code reviews, and approvals. She believes that code reviews, while important, are just simply not enough on their own. She strongly believes in production telemetry and real time monitoring of systems while they’re running. It’s not about replacing code reviews but about adding a safety net that can catch what code reviews cannot.

Lastly, ensuring that any strange patterns are looked at and monitored carefully. By consistently checking logs, monitoring dashboards, and analyzing patterns in how the system behaves, teams can catch suspicious activity early on. For example, they may notice a repeated railed login attempt, or system behaving oddly during off hours. These could be early warning signs that there is something like a security breach in the system. Teams need to make this kind of monitoring a daily habit rather than relying on auditing to do the work. That way, if something doesn’t look right, it can be taken care of right away.

In conclusion, both of these case studies provide valuable lessons for developers. Ensuring that traditional security and compliance methods aren’t just “enough” in today's fast paced tech environments. Both Bill Shinn and Mary Smith make great points about how important it is to rethink the normal approach. And instead of relying on things like code reviews or audits, developers need to build compliance and monitoring into the development process. This allows for problems to be caught and solved early on and to stay ahead of any potential threats. These things need to be a part of the daily development process from the very beginning.

Reference:

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook: How to create world-class agility, reliability, & security in technology organizations* (2nd ed.). IT Revolution.