Clay Lankford

07/25/2024

Module 12.2

**Proving Compliance in Regulated Environments**

Bill Shinn, a principal security solutions architect at Amazon Web Services, has significant experience helping large enterprise customers like Hearst Media, GE, Phillips, and Pacific Life comply with regulations while using public cloud services. He identifies a core challenge: traditional audit methods, which rely on sampling physical servers, are not suited for dynamic, code-driven DevOps environments.

Shinn notes that auditors traditionally ask for screenshots and CSV files to verify asset management, access control settings, and server logs. However, in a DevOps context where infrastructure is managed as code and servers auto-scale, these methods are impractical. To address this, Shinn's teams work closely with auditors during the control design process, using an iterative approach to define audit evidence requirements for each sprint. This collaboration ensures auditors can access necessary information on demand when services are in production.

A key strategy Shinn advocates is utilizing telemetry systems like Splunk or Kibana. These systems allow auditors to self-service their audit needs by searching for evidence directly, eliminating the need to request data samples. This approach enhances transparency and visibility, reducing the probability of errors and security flaws.

Understanding specific regulatory requirements, such as those under HIPAA, involves thorough examination of relevant legislation to identify necessary technical safeguards and audit controls. Compliance efforts must focus on preventing, detecting, and correcting issues, often through configuration settings in version control or monitoring controls.

The DevOps Audit Defense Toolkit exemplifies how organizations can document compliance processes, bridging the gap between DevOps practices and audit requirements. It outlines how to design controls within deployment pipelines to mitigate risks and demonstrates control effectiveness through control attestations and artifacts.

**Production Telemetry for ATM Systems**

Mary Smith (a pseudonym), leading the DevOps initiative for a consumer banking division of a large US financial services organization, highlights a critical observation: Information Security, auditors, and regulators often overly rely on code reviews to detect fraud. Instead, Smith argues, effective risk mitigation requires robust production monitoring controls, complemented by automated testing, code reviews, and approvals.

Smith recounts an incident from years ago where a developer planted a backdoor in the code deployed to ATM machines. This backdoor allowed the ATMs to be put into maintenance mode at specific times, enabling unauthorized cash withdrawals. Despite the presence of separation of duties between Development and Operations and a change approval process, this backdoor was not detected through code review. However, the fraud was quickly identified during a regular operations review meeting when an anomaly in the maintenance mode schedule of ATMs in a city was observed. This detection occurred even before the scheduled cash audit process.

This case study underscores that fraud can still occur despite traditional security measures such as separation of duties and change approval processes. The key to quickly identifying and mitigating such risks lies in effective production telemetry, which provides real-time visibility into system operations. Telemetry can highlight unusual patterns and activities that might indicate fraud or errors, which code reviews and traditional security practices might miss.