A Candid Look at the Shifting Landscape of Change Management for Audit

Change management is one of the fundamental parts of any enterprise IT environment. The effectiveness of change management controls has a direct impact on overall enterprise risk. In this era of fast-paced technological advancement, organizations are adopting automation to streamline their processes. For automated systems to operate effectively and achieve the desired outcomes, it is essential to have sophisticated change management controls.

Although the improper use of software engineering tools has the potential to open avenues that can be easily exploited by malicious actors, it would be a gross misstatement to say that the responsibility for change management is limited only to the engineering department. The compliance department has an obligation to educate the board and executive management about the importance of implementing a robust change management process. As the auditees, it is important for the engineering and DevOps teams to understand the effects of improper change management practices.

Auditors also play a key role when auditing artifacts related to sophisticated tools to obtain reasonable assurance of the effectiveness of new change management controls. It is essential for all audit stakeholders to learn the industry's common flaws and the latest technical controls to easily adapt to new change management audit practices.

Common Practices for Auditees

Code repository tools, continuous integration/continuous delivery (CI/CD) tools and change documentation are important building blocks of engineering change management. The actions of the auditees using these elements have a direct impact on the effectiveness of the enterprise change management practice.

Code Repositories and Continuous Integration and Deployment

Code repository tools are commonly used to store source code and control versions of code. As organizations move toward automation, it is increasingly important to use the branch rule protection settings offered by code repository and version control software. Branch rule protection settings serve as a powerful configuration to enforce segregation of duties (SoD). They enable administrators to prevent any kind of unauthorized changes from being merged to the main branch.

A common flaw is allowing engineers to modify the branch rule settings at will. This flaw nullifies the purpose of the branch rule protection settings. If allowed, an engineer could change the number-of-reviewers requirement to zero to bypass the reviewer requirement when merging changes to the main branch. This opens the potential for engineers or bad actors to act maliciously and deploy their own changes without authorization—violating the SoD requirement.

KISHAN SATHYANARAYANAN | CISA, CCSFP

Is a senior manager of third-party attestation practice at BDO LLP. He has a decade of experience in the accounting industry, including with the Big Four accounting firms. He has worked with clients in the Asia Pacific region, Europe and the United States. He specializes in information systems audit and has experience with clients in industries such as the cloud, cybersecurity, finance, entertainment, healthcare, manufacturing and blockchain technology.

With the advent of new automated software engineering tools, the responsibilities of change management auditors have changed.

Enterprises that are serious about implementing a robust change management process should encourage implementation of stronger controls to prevent engineers from making rogue changes. It is important for repository administrators to limit the number of users with the ability to modify branch rule protection settings.

CI/CD is the ongoing delivery of applications to customers with the help of automation. CI/CD introduces automation that makes the phases of change management perpetual. CI/CD brings together the development and operations teams to work in an Agile manner. CI/CD tools serve as some of the busiest pipelines of change management with continuous builds and deployments. One common practice is failing to terminate containers and virtual machines after tasks are completed. To reduce the probability of attacks, it is important to remove these tools when they are not necessary and use them in read-only mode whenever possible.

Code repositories often contain sensitive information that is used to run an enterprise. Robust and secure change management practices prevent code repositories from becoming a breeding ground for hackers. Some of the best practices used by organizations with mature change management environments include:

- Implementation of alerting tools to notify all administrators whenever a rule set is modified
- Requirement to seek the approval of another administrator before modifying rule sets for any critical purposes
- Monthly or quarterly administrative review of the code repository log to flag any unauthorized merges or changes

Change Documentation

Change tickets usually serve as the primary evidence auditors use to determine whether changes were deployed after necessary approvals were obtained and if SoD was maintained. The tickets usually serve as a transcript of the conversation between the engineers working on a specific change. The change reviewer usually logs the approval as part of the conversation in these tickets. When approvals are not explicit, the auditor must clarify the context of the conversation with the specific engineers to determine whether approval was obtained. This task takes longer when the sample size of changes tested and the number of samples with insufficient information are high. The compliance team that oversees the audits should encourage the engineering teams to improve the quality of change documentation. If the reviewer, tester (as needed) and developer deliver explicit information, the amount of time spent during audits can be significantly reduced.

Another common practice is obtaining approval for a change via an instant messaging platform. Although some organizations retain instant messaging chat logs for more than a year, others delete them after 60 to 90 days. If the chat transcripts are retained, the engineering teams can produce evidence to show the auditors from the archives. However, these approvals can get lost, or they may offer no evidence to support the change approval at the time of audit. Therefore, it is important for stakeholders to discourage the practice of obtaining approvals via instant messaging and document them in the change tickets so there is a single source of evidence. This makes the audit process more efficient and saves precious time during the audit.

Responsibilities of Auditors

With the advent of new automated software engineering tools, the responsibilities of change management auditors have changed. It is important for auditors to understand these changes and plan their audit approach as applicable, including:

- Taking a holistic approach to change management (The practice of solely relying on the population collected from the ticketing system and testing those tickets may not be sufficient.)
- Understanding the chain of the process (i.e., development, test, deployment and rollbacks)
- Understanding the review requirements of each repository in scope and the automated controls in place to enforce the segregation of duties

LOOKING FOR MORE?

- Read *IT Audit Fundamentals Study Guide*. *www.isaca.org/it-audit-fundamentals-study-guide*
- Learn more about, discuss and collaborate on audit and assurance in ISACA's Online Forums. *https://engage.isaca.org/onlineforums*

- Examining the list of users who have developer and administrator access to each repository in scope (This will also help clarify which users can modify the rule sets of each repository.)
- Learning which users have access to the production environment and determining which have privileged access to both the production environment and deployment tools, plus the code repository tools (It may not always be a red flag to see a user with privileged access to both the code repository and the production environment because the DevOps teams have multidisciplinary skill sets and they work on a continuous-loop model to manage the application life cycle through the development, test, deploy, operate and repeat phases.)
- Studying the activity log generated from the code repository tools (If any suspicious activities or rogue merge or modifications of automated rule sets are noted in the log, necessary due diligence should be undertaken to understand the reasons behind these activities.)
- Studying the log of all users added and removed to the code repository tools and production and deployment tools during the audit period (The list of users obtained at a point in time may not give a clear picture of whether any unauthorized users gained access to critical tools during the audit, even for a short period of time, because it only captures new users and not removed users or temporary users.)
- Determining the best source to generate the population of changes that were deployed to the production environment (Every ticket entry of a population generated from a general ticketing system may not have impacted the production environment. It is important to figure out the best source to get the complete and accurate list of changes that impacted the production environment.)
- Remembering the traditional requirement of maintaining SoD for any changes that reach the production environment (At a minimum, more than one person should be aware of the changes deployed to the production environment. In an era of automated deployments, it is critical for auditors to understand how the changes are reviewed and merged to the main branch.)
- Ensuring that the production data are not used in any form whatsoever in the test or development environment
- Performing the traditional testing of access provisioning, access deprovisioning and user and admin access review testing despite all other due diligence, such as examination of logs and point-in-time user listings

Mature change management practices help enterprises better position themselves to face new regulations and reduce the risk of future security threats.

Conclusion

Change management is a dynamic and evolving part of audit and is therefore one of the highest risk areas of an audit. The approach taken by an auditor to address change management risk should be less about checking boxes and more about using professional judgement. The range of engineering tools used by each organization is different, so it is essential for auditors to learn the features of new software engineering tools in a timely manner and frame their change management audit approaches accordingly. The knowledge developed helps auditors evaluate whether a meaningful level of assurance has been obtained while auditing the artifacts related to new types of change management controls. At the same time, enterprises should focus on maturing their change management practices rather than simply passing an audit. Mature change management practices help enterprises better position themselves to face new regulations and reduce the risk of future security threats.