15 IT Audit Risks and Tactical Mitigation Strategies

graylog

Table of Contents

Executive Summary

"Ping!" You open your email and see the vaguely dreaded email with a list of audit documentation. With the annual IT audit on the horizon, you start going back through the list of security controls and sort through the reports that you've been sending to leadership for the last ten months. Even with all the documentation, you have that little voice in the back of your head telling you to double check for potential risks one more time.

The annual information technology (IT) audit reviews an organization's systems, data flows, and third-party integrations to ensure that controls function as intended. In today's connected world, organizations use their IT audits to give customers assurance over their resilience and security.

To prepare for an upcoming audit, organizations should review these 25 tactical audit risks and mitigate them as quickly as possible.

What is audit risk?

Audit risk is the possibility that an auditor will find an issue with an organization's financial statements or operational effectiveness. More specifically, IT audit risk refers to potential errors, fraud, or compliance violations with the IT environment that could lead to material misstatements, operational disruptions, or regulatory sanctions.

What are the types of audit risk?

IT audit risks are the potential that a system or process could lead to a material misstatement, error, or fraud, typically falling into several categories:

- **Inherent risk:** An unfiltered risk arising from the IT environment.
- **Control risk:** Failure of internal controls, like policies, procedures, or practices.
- **Detection risk:** Failure of the audit process to identify any issues, like sampling limitations.

Identity and Access IT Audit Risks

Modern cloud environments must limit user access to resources according to the principle of least privilege and monitor for potential unauthorized access.

Incomplete offboarding

With complex environments, organizations often lose visibility into access to secondary systems, across APIs, and within shared resources. **Typically, the audit test may look at samples like:**

- The last 30-60 days of terminations.
- 5-10 key systems and their active access.
- Active sessions, tokens, or shared access.

Some best practices for mitigating risk include:

- Tying offboarding to an identity provider or human resources system.
- Creating a way to automatically deprovision integrated applications.
- Collecting evidence for non-integrated systems.

As part of improving the offboarding process, organizations can consider:

- Building an inventory for critical applications and an offboarding checklist for them.
- Running a weekly terminated user audit.
- Revoking or rotating any risky API keys.

Overprovisioned access

Over time, organizations often find that users have more access than they need to complete their job roles. However, identifying this excess access is difficult. **Typically, an audit test may look at samples like:**

- Users with access to critical systems.
- Difference between access granted and job function requirements.
- Administrative privileges granted to ensure no one has unnecessary access.

Some best practices for mitigating risk include:

- Aligning role definitions with least privilege requirements.
- Regularly reviewing access, typically every financial quarter.
- Providing clear justification for elevated access.

As part of improving processes, organizations can consider:

- Removing obvious excess administrative access.
- Creating a limited number of baseline roles to make granting and reviewing more management.
- Implementing timebound access for privileged roles.

Lack of service account ownership

Service accounts and API keys are machine identities that organizations often fail to tie to a human user identity. Without linking the two, organizations struggle with accountability for and governance over them. **Typically, an audit test may look at samples like:**

- An exported list of service accounts and API keys to identify accounts without owners or descriptions.
- Long-lived or non-rotated keys that exceed the defined rotation policies.
- Active applications or workflows to identify dormant or potentially abandoned service accounts and keys.

Some best practices for mitigating risk include:

- Mapping all accounts to an owner and use case.
- Creating an inventory for machine identities.
- Regularly reviewing keys and rotating them.

As part of improving processes, organizations can consider:

- Disabling unknown or unused accounts.
- Creating alerts when someone creates a new service account without assigning or tagging an owner.
- Establishing and enforcing a standardized workflow for provisioning service accounts.

Stale accounts

Stale accounts are inactive accounts that remain enabled. Organizations often struggle to identify these when no one clearly owns the monitoring. **Typically, an audit test may look at samples like:**

- Accounts with no login activity during the identified sampling period.
- Accounts that remain inactive after employee termination dates or human resource exit events.
- Accounts without recent authentication logs according to the identity and access management tool.

Some best practices for mitigating risk include:

- Automatically disabling inactive accounts.
- Periodically cleaning up access.
- Monitoring for dormant access.

As part of improving processes, organizations can consider:

- Setting inactivity thresholds for automatically disabling access.
- Mapping inactivity detention across systems for consistent enforcement.
- Correlating account activity with device or session logs to identify accounts without meaningful user interaction.

Multi-Factor Authentication (MFA) gaps

Implementing MFA is a critical control to mitigate account takeover and credential-based attack risks. However, organizations often struggle with MFA as the process creates extra work for users and legacy systems may make adoption difficult. **Typically, an audit test may look at samples like:**

- User accounts across critical systems to verify MFA enrollment and enforcement.
- Privileged or administrative accounts to identify any MFA exemptions or inconsistencies.
- Remove access points to confirm MFA implementation and enforcement.

Some best practices for mitigating risks include:

- Enforcing MFA across all users, especially ones with privileged access.
- Minimal exceptions with documented reasoning for each one.
- Centralized enforcement using an identity provider.

As part of improving processes, organizations can consider:

- Testing MFA bypass scenarios to identify potential gaps, like across legacy protocols or service exceptions.
- Regularly auditing MFA logs to validate enforcement and identify push fatigue patterns.
- Periodically validating MFA enforcement at the policy level to identify potential issues with new systems and integrations.

Systems and Asset Management IT Audit Risks

Without appropriate systems and asset management, organizations may lack visibility into and control over their environment which leads to risks arising from vulnerabilities, misconfigurations, and unauthorized devices or software.

Unknown assets ("shadow IT")

Shadow IT are devices or systems that exist outside the organization's official asset inventory. When organizations lack a comprehensive inventory, they are unable to secure these assets. **Typically, an audit test may compare the asset inventory against data from samples like:**

- Data generated by Mobile Device Management (MDM), Endpoint Detection and Response (EDR), or cloud platforms.
- Network scans or logs to detect devices communicating on the network.
- List of Software-as-a-Service (SaaS) applications or cloud resources used across the organization and the approved vendor list.

Some best practices for mitigating risks include:

- Creating a single, centralized asset inventory.
- Engaging in ongoing, continuous asset discovery.
- Assigning an owner to each asset.

As part of improving processes, organizations can consider:

- Integrating asset inventory updates into procurement and onboarding workflows to capture new assets during creation.
- Correlating asset data across tools to automatically reconcile discrepancies or identify gaps.
- Establishing periodic periodic reconciliation checks between network activity and inventory to continuously validate accuracy.

Untracked asset life cycle

Often, organizations struggle to update asset inventories consistently, including when retiring them. **Typically, an audit test may look at samples like:**

- Assets marked retired or disposed to verify that they are no longer marked active in other tools, like identity systems or MDM.
- Inactive assets to confirm whether they are properly decommissioned.
- Procurement or disposal records to compare against the asset inventory to identify discrepancies.

Some best practices for mitigating risks include:

- Creating processes to track assets from procurement through disposal.
- Establishing a formal decommissioning process.
- Wiping data from and revoking access to all decommissioned assets.

As part of improving processes, organizations can consider:

- Requiring formal verification before marking assets as decommissioned to ensure governance across the lifecycle.
- Establishing automation for decommissioning triggers, like data generated by human resources, procurement, or inactivity signals.
- Linking asset records to identity and access systems for consistency when revoking access.

Unpatched systems

Many organizations struggle to apply security updates to systems, especially as they onboard new assets and the number of vulnerabilities discovered increases year over year. Attackers use these unpatched systems as an initial access vector for compromising systems. **Typically, an audit test may look at samples like logs generated by:**

- Endpoint management and patching tools to validate patch status, identify missing updates, and review compliance on a device level basis.
- Vulnerability scanning tools to identify missing patches for known vulnerabilities.
- Asset inventory or configuration management database to ensure coverage across all environments, including production, development, endpoints, and servers.

Some best practices for mitigating risk include:

- Basing remediation timeline on vulnerability criticality, asset important, and known exploit activity.
- Automating patch pipelines for critical systems.
- Documenting exceptions and compensating controls when not automating the update process.

As part of improving processes, organizations can consider:

- Tracking remediation by integrating vulnerability findings into ticketing and change management workflows.
- Establishing a defined "patch validation" step to confirm updates actually remediate the vulnerability.
- Prioritizing patching efforts based on attack surface and segmenting systems by exposure, like internet-facing, internal, and restricted.

Misconfigured non-production environments

Some organizations treat non-production environments as low risk, leaving these development and testing environments with weaker controls despite the sensitive data that they contain. **Typically, an audit test may look at samples like:**

- Documentation of both production and non-production environments to identify discrepancies around access controls, authentication requirements, and privilege levels.
- Non-production datasets to identify any sensitive customer information or confidential data used in development, testing, or staging systems.
- User access in non-production environments to identify any overly broad access.

Some best practices for mitigating risk include:

- Maintaining separate production and non-production environments by enforcing logical network and identity boundaries.
- Using synthetic or masked data in non-production environments.
- Ensuring non-production has the same baseline security controls as production.

As part of improving processes, organizations can consider:

- Implementing automated, continuous monitoring to detect potential data leaking from production environments.
- Explicitly classifying, reviewing, and approving data sets before replicating them in development or testing systems.
- Establishing appropriate governance to enforce risk-based controls across development, testing, staging, and production.

Monitoring and Detection IT Audit Risks

Monitoring and detection capabilities signal an organization's ability to identify and investigate security incidents, preventing small issues from escalating into major data breaches that can disrupt business, undermine customer trust, and impact financial stability.

Failure to investigate alerts

Many security teams struggle with alert fatigue, high volumes of false positives that have them investigating irrelevant issues. However, this can lead to missing important alerts and allowing real security incidents to remain uninvestigated, increasing attacker dwell time and the potential damage. **Typically, an audit test may look at samples like:**

- High-severity alerts to verify that the team reviewed, investigated, and responded appropriately.
- Alert backlog to identify any alerts that remain outstanding beyond defined triage and response time thresholds.
- False positive rate and alert tuning history to assess rule maintenance and refinement.

Some best practices for mitigating risk include:

- Establishing alert prioritization tiers for high-fidelity and high-impact alerts.
- Defining ownership and escalation paths for each alert tier.
- Using investigation outcomes to tune detection rules and reduce false positives.

As part of improving processes, organizations can consider:

- Regularly reviewing alerts so teams can actively retire, merge, or adjust detections.
- Enriching alerts with context like asset criticality, user risk, or threat intelligence.
- Tracking alert-to-action ratios to identify which detections consistently generate value while reworking or retiring low-value alerts.

Logging gaps

Logging gaps often arise from latency, fragmentation, and ingestion limits that can drop or delay information about events. Even more challenging, different technologies across an environment generate data in various formats, leading to consistency and correlation issues. **Typically, an audit may look at the following samples:**

- Logs from key systems to confirm that they consistently capture all security-relevant events, like authentication, privilege changes, and access to sensitive data.
- Logs across multiple systems to ensure end-to-end correlation, like logs across identity, endpoint, and application.
- Log ingestion and retention settings to identify delayed, incomplete, or insufficient log availability.

Some best practices for mitigating risk include:

- Normalizing log formats and schemas across systems for consistent parsing and cross-system correlation.
- Centralizing log collection through a unified ingestion pipeline.
- Defining minimum logging requirements across all environments for critical event types, like authentication, access, and privilege changes.

As part of improving processes, organizations can consider:

- Implementing tiered ingestion architecture to prioritize and preserve high-value security events during periods of high log volume.
- Automating validation checks to detect broken log pipelines, schema drift, or missing event sources in near real time.
- Aligning logging configurations with system criticality so that high-risk applications generate richer, more detailed telemetry by default.

Lack of central visibility

As organizations adopt more IT and security tools, logging and monitoring becomes fragmented and siloed while integration becomes more complex. Without central visibility, organizations may lose sight of security and operational incidents, making remediation and debugging more time-consuming and impacting SLAs. Typically, an audit may look at the following samples:

- List of in-scope systems and data sources that feed into the monitoring and logging architecture.
- Evidence showing log ingestion and retention configurations across key systems.
- Sample logs or incident records that demonstrate correlations across multiple systems.

Some best practices for mitigating risk include:

- Using centralized location for maintaining an up-to-date inventory of log sources, including system ownership and logs generated.
- Implementing change control requirements to ensure responsible parties review and update logging configurations during any system or application change.
- Documenting and testing incident investigation procedures to ensure forensic and operational analysis capabilities.

As part of improving processes, organizations can consider:

- Implementing automated alerting for log degradation indicators, like sudden drops in event volume or ingestion lag.
- Ensuring that the onboard process for new systems includes logging integration prior to production deployment.
- Periodically testing log usability during incident simulation to confirm that data supports real investigation workflows.

Change and Configuration Management

Change and configuration management ensures governance over updates that mitigates risks related to misconfigurations, security gaps, and system instability.

Untracked changes

When system changes occur without documentation or approval, organizations can face security risks and operational disruptions. Organizations often struggle with documenting these changes when they need to prioritize speed and availability over process. **Typically, an audit may look at the following samples:**

- Tickets or logs showing change management approvals, implementation details, and timestamps for system and application changes.
- System configuration baselines and version history records documenting expected versus current configurations.
- Deployment logs or release records from CI/CD or change management tools showing what was changed, when, and by whom.

Some best practices for mitigating risk include:

- Implementing and enforcing a formal workflow that traces all changes while capturing approval, implementation details, and rollback plans.
- Implementing version-controlled configuration baselines to create a source of truth for system state.
- Embedding change validation into deployment processes to test and verify all changes prior to production.

As part of improving processes, organizations can consider:

- Standardizing change categorization to implement and consistently apply the necessary review and approval for each risk level, like standard, emergy, or high-risk.
- Automating change capture by integrating deployment tools and change management systems.
- Introducing post-implementation reviews to ensure that the outcome of high-risk changes match the approved change intent.

Unauthorized configuration drift

Undocumented changes, emergency fixes, and routine updates can cause systems to gradually shift away from their approved configuration baselines, creating a gap between assumed and actual states that can harm security and compliance. **Typically, an audit may look at the following samples:**

- Infrastructure-as-code repository history to review configuration changes over time, like Git commit logs and pull requests.
- Cloud provider configuration snapshots to review state at defined intervals.
- Exceptions or waivers that explain the justification, scope, and expiration dates for all approved deviations from standard configuration baselines.

Some best practices for mitigating risk include:

- Ensuring consistency across environments with a single source of truth for system configurations, like version-controlled templates.
- Automating provisioning and configuration updates to minimize human error risk.
- Continuously monitoring for and validating production environment configurations to detect and remediate drift.
- prior to production.

As part of improving processes, organizations can consider:

- Implementing automated reconciliation workflows that compare system state against defined configuration standards.
- Building dashboards in a centralized source of truth that aggregate configuration states across environments and trigger alerts for drift detection.
- Scheduling routine formal reviews that compare baseline definitions and update documentation to reflect approved changes.

Temporary exceptions

Often, organizations allow for exceptions to their policies responding to operational needs, technical constraints, or delivery pressure. However, when they fail to track and review the exception over time, they may have unmanaged deviations that create compliance, security, and operational risks. **Typically, an audit may look at the following samples:**

- Risk acceptance and policy exception entries for approved deviations, documenting justification, scope, approver, and defined review or expiry dates.
- Documentation of periodic reviews, including meeting minutes or attestation logs confirming the exceptions' ongoing validity.
- Approvals or workflow records formally documenting the request, assessment, and authorization processes.

Some best practices for mitigating risk include:

- Clearly and narrowly defining exception criteria so they only exist in cases when no compliant alternative is available.
- Implementing a centralized exception management process that requires all policy deviations to be formally recorded, assessed, and time-bound.
- Assigning clear system or process ownership for each exception to ensure accountability for and governance over monitoring, risk, and remediation planning.

As part of improving processes, organizations can consider:

- Embedding exception approval into standard change and deployment workflows to capture deviations when introduced to the system.
- Automating tracking and alerting for exception expiry data to ensure approvals remain up-to-date.
- Aggregating exception data into centralized reporting dashboards to identify partners of recurring deviations and address underlying control gaps.

Graylog: The Audit Control Management Solution for Lean Teams

Graylog provides a practical way to gain visibility and control over identity, access, configuration, and operational risks without the overhead of traditional enterprise tooling. By unifying audit-relevant data across systems, enabling structured detection of control gaps, and supporting continuous validation of core security processes, it helps organizations move from reactive, point-in-time auditing to an ongoing, evidence-driven approach to risk management.

Designed for resource-constrained teams operating in complex hybrid and cloud environments, Graylog reduces the manual effort required to surface issues like stale access, excessive privileges, misconfigurations, and monitoring gaps. With centralized observability, flexible deployment options, and a focus on actionable insight rather than raw log volume, it enables security and audit teams to identify control failures faster, close gaps more efficiently, and maintain stronger assurance over their environment without adding operational burden or unnecessary complexity.

ABOUT GRAYLOG

Graylog is the no-nonsense SIEM that cuts through noise and complexity. It delivers what security teams need most: full visibility, faster investigations, and smarter detection—without trade-offs or surprise costs. From automated workflows to correlation and anomaly detection, Graylog helps analysts move faster and stay focused. With a product suite spanning Graylog Enterprise, Security, API Security, and Open, we support everyone from large enterprises to lean teams. Trusted by over 60,000 organizations around the world. Learn more at graylog.com or connect with us on Bluesky and LinkedIn.

www.graylog.org | info@graylog.com

graylog