**Handling Segregation of Duties (SOD) Violations**

SOD is all about ensuring no single user has too much power within a system, which could lead to fraud or errors.

1. **What are SOD violations?**

A SOD Violations occurs when:

* A user is assigned conflicting roles and permissions.
* This conflict creates a risk of unauthorized actions.

Example:

* In a financial system:
  + A user with “**Create Invoice**” permission should not have “**Approve Invoice**” permission.
  + Why? They could create and approve fraudulent invoices.

1. **Preventing SOD Violations**

IAM/IGA tools like Saviynt have built-in features to prevent SOD Violations.

1. SOD Rules:
   1. Define rules to detect and block conflicting permissions.
   2. Example:
   3. Rule: no user should have both ‘create invoice’ and ‘approve invoice’ permissions
2. Role Design:
   1. Ensure roles are designed with least privilege principle.
   2. Example: create separate roles: “invoice creator” and “Invoice approver”.
3. Prevention During Provisioning:
   1. Use tools like Saviynt to validate permissions during onboarding or role changes.
   2. Example: When a user is assigned a new role, the system checks if it conflicts with existing roles.
4. **Identifying SOD Violations**

Use **SOD Analytics** in tools like Saviynt to:

* + Scan existing user permissions.
  + Identify users with conflicting roles or permissions.

Example:

A report shows:

* + User A has access to both "Add New Vendor" and "Approve Vendor Payments."
  + This is flagged as an SOD violation.

1. **Remediating SOD Violations**

When a violation is detected:

1. **Review Access**:
   * Analyze the user’s permissions and determine the severity of the risk.
2. **Revoke Conflicting Permissions:**
   * Work with the user’s manager to decide which permissions to revoke.
3. **Implement Compensating Controls:**
   * If permissions cannot be fully revoked:
     + Add extra monitoring (e.g., logs, alerts).
     + Require multi-level approvals for risky actions.
4. **Continuous Monitoring**
   * Use **real-time alerts** to detect potential SOD violations as they happen.
   * Schedule periodic SOD audits to review permissions and roles.

**Activity 6: SOD Violation Scenario**

Imagine you work for a company using Saviynt. You discover that:

1. **User B** is a Finance Analyst with "Create Purchase Orders" and "Approve Purchase Orders" permissions.
2. This is flagged as an SOD violation.

Tasks:

1. What steps would you take to remediate this violation?
2. How would you prevent such violations in the future?

Solution:

1. Remediation of the SOD Violation
2. **Review Access**:
   * Analyze User B’s permissions and assess the severity.
   * Investigate if User B has actually used both permissions together.
   * Check if any unauthorized actions occurred.
3. **Revoke Conflicting Access**:
   * Work with User B’s manager to revoke either "Create Purchase Orders" or "Approve Purchase Orders."
   * Ensure revocation is logged and documented for audits.
   * Update User B’s roles to reflect the change.
4. **Implement Compensating Controls**:
   * Add monitoring (logs, alerts) and multi-level approvals if permissions cannot be fully revoked.
   * Set up alerts for suspicious activity (e.g., User B creating and approving the same purchase order).
   * Require an independent auditor to review User B’s actions periodically.
5. **Prevention of Future SOD Violations**
   1. **Automated SOD Rule Enforcement**:

Define SOD rules in Saviynt to automatically detect and block conflicting roles during provisioning.

* 1. **SOD Risk Reports**:

Schedule automated reports highlighting potential SOD conflicts. Example: Generate a monthly list of users with access to both "Create" and "Approve" actions.

* 1. **Role Design and Optimization**:

Perform a **role cleanup:**

* + - Simplify roles to avoid overlaps.
    - Remove redundant or unused permissions.
  1. **SOD Awareness Training**:

Train managers and role owners to understand SOD risks and how to handle conflicts.