**# SECURE-CODE**

**Financial Access Risk Analysis - Secure Code Hackathon**

**Project Overview**

The **Financial Access Risk Analysis** project is designed to automate the detection of **Segregation of Duties (SoD)** violations in ERP systems, specifically focusing on **Oracle Fusion**. The solution processes ERP system data and generates detailed reports highlighting conflicting roles and entitlements, helping organizations reduce the risk of fraud, errors, and unauthorized access.

**Key Features:**

* Automated detection of SoD conflicts.
* Deterministic algorithm ensuring consistent results.
* Dockerized solution for portability and reproducibility.
* Efficient handling of large datasets with optimized performance.
* Easy-to-interpret output reports for compliance teams.

This solution processes user-role mappings, privilege assignments, and SoD rules using optimized role-based access control (RBAC) logic and high-performance data structures. The implementation is fully Dockerized, allowing seamless execution with no modifications required by the evaluator.

## ****Core Concepts****

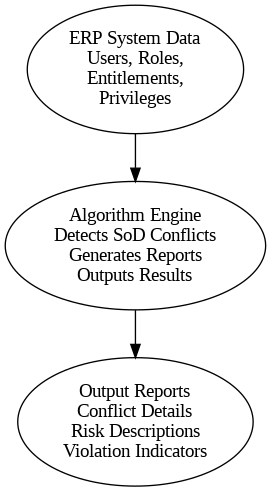
### **Segregation of Duties (SoD)**

SoD ensures that no individual can perform both **critical actions** in a business process, reducing the risk of fraud and unauthorized activities.  
Example:

* **Leg 1:** Create Payables Invoices
* **Leg 2:** Approve Payables Invoices  
  If the same user has both entitlements, it creates a **conflict** and raises a security red flag.

### **Key Data Components**

1. **Entitlements:**
   * Permissions granted to users, groups, or roles to access specific features.
   * Stored in XX\_FUSION\_SOD\_MST and XX\_FUSION\_9\_ENT\_MST tables.
2. **Privileges:**
   * Specific actions a user can perform, bundled into duty roles.
   * Stored in XX\_FUSION\_7\_PVLGS\_MASTER and mapped using XX\_FUSION\_6\_PVLG\_TO\_ROLE.
3. **Roles:**
   * Groups of privileges assigned to users.
   * Defined in XX\_FUSION\_4\_ROLE\_MASTER\_DETAILS and mapped through XX\_FUSION\_3\_USER\_ROLE\_MAPPING.
4. **Users:**
   * Individuals with access rights based on assigned roles.
   * Stored in XX\_FUSION\_2\_USER\_DETAILS



**Features**

Deterministic Algorithm: Produces the same output for the same input, ensuring consistency.

Optimized RBAC Engine: Efficiently expands role hierarchies and privilege mappings.

Fast Conflict Detection: Uses hash-based lookup tables to speed up SoD violation detection.

Dockerized Execution: The solution is containerized to ensure portability across environments.

Setup and Execution Instructions

3.1 Prerequisites

Before running the solution, ensure you have the following installed:

* Docker Desktop (Windows, Linux, or macOS)
* Git installed for cloning the repository.

Docker ([Install from Docker Official Site](https://docs.docker.com/engine/install/))

Excel Data Files (Provided by the competition organizers)

### **Building and Running the Docker Container**

#### **Cloning the Repository**

bash

Copy code:

git clone https://github.com/DEBDEEP-BANERJ2E/Secure\_Code

cd Secure\_Code

#### **Building the Docker Image**

bash

Copy code

#### docker pull debdeep3613peed/secure-code:latest

#### **Running the Container**

1. Prepare your test input in a directory named /input.
2. Run the Docker container with volume mapping:

bash

Copy code

docker run -v "/path/to/input:/input" -v "/path/to/output:/output" financial-access-risk-analysis /input/test\_case.xlsx /output/results.xlsx

* /input: Directory containing the input Excel/CSV files.
* /output: Directory where the results will be stored.

## 🛠️ ****Dependencies****

### 🔥 **Inside Docker**

* **Programming Language:** Java / C++
* **Database:** Oracle ERP data tables
* **Libraries:**
  + Apache POI: For Excel file processing.
  + OpenCSV: For CSV file handling.
  + Log4j: For logging operations.
* **Docker Image:**
  + Base image: openjdk:11-jre-slim (Java 11 runtime)
  + Oracle Database client libraries for table interactions.
  + Output stored in .xlsx or .csv format.

## ****Security and Compliance****

* Ensures compliance with **Oracle Fusion** security policies.
* Prevents **unauthorized access** by identifying SoD violations.
* Improves **audit readiness** with detailed reports.

## ****License****

This project is licensed under the MIT License. See the <LICENSE> file for details

TEAM DR  
(DEBDEEP BANERJEE,ROHAN JAIN)