PRJ-CRISC-050: Quantitative Risk Scenario Modeling

Certification: Certified in Risk and Information Systems Control (CRISC) **Domain:** Domain 2 - IT Risk Assessment

---

1. Project Overview

This project operationalizes a core component of the CRISC certification: the development and analysis of risk scenarios. Instead of relying solely on qualitative (High, Medium, Low) assessments, this project demonstrates how to build a quantitative, data-driven model to assess IT risk. We will create a system that ingests various data sources, applies a simplified risk model to calculate a quantitative risk score, and visualizes the results to support risk-based decision-making.

We will define a specific risk scenario, such as: “A public-facing web server with an unpatched vulnerability is compromised by an external attacker, leading to a data breach.” Our system will then programmatically assess the **likelihood** of this scenario (by checking for unpatched, public servers) and the potential **impact** (by classifying the data on the server). The resulting risk scores will be stored in a risk register and visualized on a dashboard, enabling stakeholders to prioritize remediation efforts based on clear, quantitative data.

Key Objectives

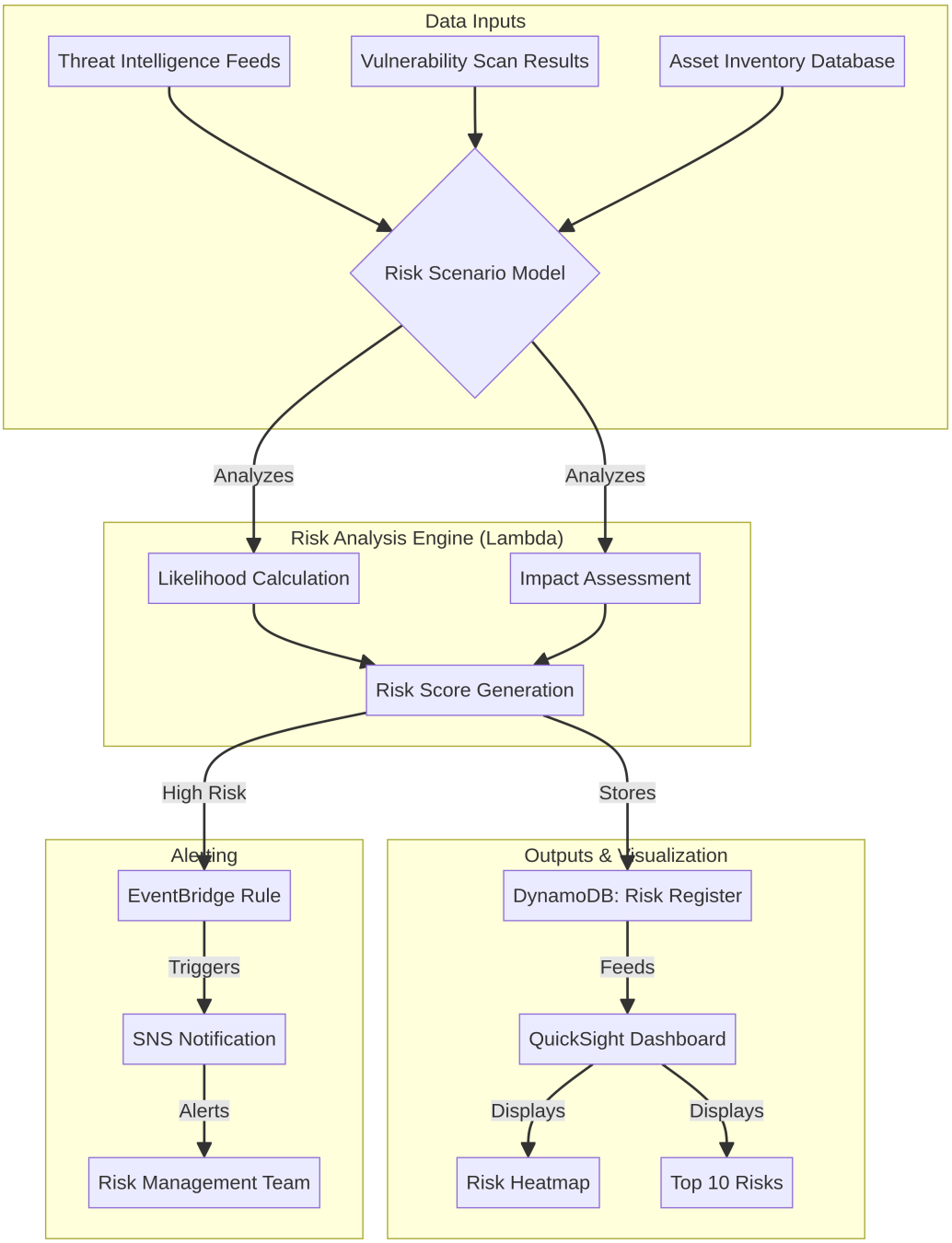
- Understand the process of developing and analyzing IT risk scenarios.
- Differentiate between qualitative and quantitative risk assessment.
- Build a serverless application to automate the risk assessment process.
- Ingest data from multiple sources (e.g., vulnerability scanners, asset inventories) to inform the risk model.

- Develop a simplified model to calculate risk likelihood and impact, resulting in a quantitative risk score.
- Store the results in a **DynamoDB** table, creating a dynamic risk register.
- Visualize the risk landscape using **Amazon QuickSight** to create risk heatmaps and dashboards.

---

2. Architecture

The architecture is a serverless data pipeline that feeds a risk analysis engine, which in turn populates a risk register and a visualization dashboard.



Architectural Flow:

1. **Data Inputs:**
    - The system is designed to ingest data from various sources that provide context for risk assessment. For this project, we will simulate these inputs:
        - **Threat Intelligence Feeds:** Information about active exploits and attacker tactics.
        - **Vulnerability Scan Results:** Data from tools like Amazon Inspector or Tenable, identifying CVEs on specific assets.

        - **Asset Inventory Database:** A CMDB that contains information about each asset, including its owner, business criticality, and the classification of data it stores.

2. **Risk Analysis Engine:**
    - A scheduled **EventBridge Rule** triggers an **AWS Lambda function** (`Risk-Analysis-Engine`).
    - The Lambda function executes the **Risk Scenario Model**. It fetches data from the various input sources.
    - **Likelihood Calculation:** The model assesses the likelihood of the scenario. For our example, it would check: "Are there any servers with `public_ip=True`, `vulnerability=CVE-XXXX`, and `threat_intelligence_indicates_active_exploit=True`?"
    - **Impact Assessment:** The model assesses the potential impact. It would look up the asset in the inventory and check: "What is the `data_classification` of the data on this server?" (e.g., Public, Confidential, Highly Confidential). This is then mapped to a financial value.
    - **Risk Score Generation:** The final risk score is calculated, often as `Risk = Likelihood * Impact`.

3. **Risk Register and Visualization:**
    - The calculated risk score, along with all the contextual data, is written to a **DynamoDB table**, which serves as our live **Risk Register**.
    - **Amazon QuickSight** is connected to the DynamoDB table. It reads the risk register data and visualizes it.
    - The QuickSight **Dashboard** can include:
        - A **Risk Heatmap** (a scatter plot of Likelihood vs. Impact).
        - A ranked list of the **Top 10 Risks**.
        - Trends of specific risks over time.

4. **Alerting:**
    - An **EventBridge Rule** can be configured to listen for risk scores that exceed a certain threshold. When a high-risk item is written to DynamoDB, the rule

triggers an **SNS Notification** to alert the **Risk Management Team** for immediate attention.

---

3. Prerequisites

- An AWS account with administrative permissions.
- An Amazon QuickSight subscription (Standard or Enterprise).
- Sample data (e.g., CSV files) representing your asset inventory and vulnerability scans.

---

4. Step-by-Step Implementation Guide

Step 4.1: Set Up the Data Stores

1. **Create DynamoDB Tables:**
    - Go to the **DynamoDB Console -> Create table**.
    - **Table 1:** **`AssetInventory`** : Partition key: `AssetID` (String). Add sample items representing your servers, including attributes like `BusinessCriticality` and `DataClassification`.
    - **Table 2:** **`VulnerabilityScans`** : Partition key: `AssetID` (String), Sort key: `CVE` (String). Add sample items representing vulnerabilities found on your assets.
    - **Table 3:** **`RiskRegister`** : Partition key: `RiskID` (String). This table will be populated by our Lambda function.

Step 4.2: Create the Risk Analysis Lambda Function

1. **Create an IAM Role:** Create a role for the Lambda function with permissions to read from the `AssetInventory` and `VulnerabilityScans` tables and write to the `RiskRegister` table.
2. **Create the Lambda Function:**

- **Name:** `Risk-Analysis-Engine`
- **Runtime:** Python 3.9
- **Role:** Choose the IAM role you created.
- **Code:** Paste the following conceptual Python code. This script outlines the logic for the risk calculation.

```
import boto3
import uuid


def calculate_likelihood(vulnerability):
    # In a real model, this would be more complex
    # e.g., based on CVSS score, threat intel, etc.
    if vulnerability["CVSS_Score"] > 7.0:
        return 0.8 # High likelihood
    return 0.3 # Medium likelihood


def calculate_impact(asset):
    # Map data classification to a financial value
    impact_map = {
        "Public": 1000,
        "Internal": 50000,
        "Confidential": 250000,
        "Highly Confidential": 1000000
    }
    return impact_map.get(asset["DataClassification"], 0)


def lambda_handler(event, context):
    dynamodb = boto3.resource("dynamodb")
    asset_table = dynamodb.Table("AssetInventory")
    vuln_table = dynamodb.Table("VulnerabilityScans")
    risk_table = dynamodb.Table("RiskRegister")


    # Iterate through all vulnerabilities
    for vuln in vuln_table.scan()["Items"]:
        asset = asset_table.get_item(Key={"AssetID": vuln["AssetID"]})
["Item"]


        likelihood = calculate_likelihood(vuln)
        impact = calculate_impact(asset)
        risk_score = likelihood * impact


        # Write to the risk register
        risk_table.put_item(
            Item={
                "RiskID": str(uuid.uuid4()),
```

```
                "Scenario": "Data breach via unpatched vulnerability",
                "AssetID": asset["AssetID"],
                "Likelihood": str(likelihood),
                "Impact": str(impact),
                "RiskScore": str(risk_score)
            }
        )
    return {"status": "SUCCESS"}
```

Step 4.3: Schedule the Lambda with EventBridge

1. Create an **EventBridge Rule** to trigger the `Risk-Analysis-Engine` Lambda function on a schedule (e.g., daily).

Step 4.4: Visualize the Data in QuickSight

1. **Grant QuickSight Access to DynamoDB:** Ensure your QuickSight service role has permissions to read the `RiskRegister` DynamoDB table.
2. **Create a New Dataset:**
    - In QuickSight, go to **Datasets -> New dataset**.
    - Choose **DynamoDB**.
    - **Data source name:** `RiskRegisterSource`
    - Select the `RiskRegister` table and click **Select**.
    - Choose **Directly query your data** and finish.
3. **Create an Analysis and Dashboard:**
    - Create a new analysis using the `RiskRegister` dataset.
    - **Create a Heatmap:**
        - Choose a **Scatter Plot** visual.
        - Drag `Likelihood` to the X-axis, `Impact` to the Y-axis, and `RiskScore` to the Size field.
        - This creates a classic risk heatmap.
    - **Create a Top Risks Table:**
        - Add a **Table** visual and populate it with `RiskID`, `AssetID`, and `RiskScore`.

        - Sort the table by `RiskScore` in descending order.
    - Publish the analysis as a **Dashboard** named `IT Risk Dashboard`.

---

5. The CRISC Perspective

This project directly supports the tasks of a risk professional:

- **Risk Identification:** The system identifies specific, concrete risk scenarios affecting individual assets.
- **Risk Analysis:** It moves beyond qualitative guesses to a quantitative analysis of likelihood and impact, based on real data.
- **Risk Evaluation:** The dashboard allows stakeholders to evaluate the overall risk posture and compare different risks, enabling informed decisions about where to allocate resources.
- **Risk Monitoring:** The automated, scheduled nature of the system ensures that the risk register is not a static document but a living, continuously updated view of the organization's risk landscape.

---

6. Cleanup

1. **Delete the QuickSight dashboard and analysis.**
2. **Delete the dataset** in QuickSight.
3. **Delete the EventBridge rule.**
4. **Delete the Lambda function.**
5. **Delete the IAM role** for the Lambda function.
6. **Delete the three DynamoDB tables.**

Business Context

The Problem

Organizations lack visibility into cloud security risks and cannot prioritize remediation efforts effectively. Risk assessments are manual, infrequent, and outdated. Security teams struggle to quantify risk and communicate it to business stakeholders.

The Solution

Automated risk assessment and management platform for AWS environments. Continuously identifies, analyzes, and prioritizes security risks based on business impact. Provides risk scoring, trend analysis, and executive reporting. Integrates with remediation workflows for risk mitigation.

Business Value

- **Risk Prioritization:** Focus on high-impact risks first, improving ROI of security efforts
- **Business Alignment:** Risk scores tied to business impact, not just technical severity
- **Continuous Assessment:** Real-time risk visibility vs. annual assessments
- **Executive Communication:** Risk dashboards translate technical issues to business language

Risk Mitigation

Identifies critical risks before exploitation, prioritizes remediation efforts, reduces overall risk exposure, and ensures alignment with business risk appetite.

GRC Mapping

Compliance Frameworks

- **ISO 31000:** Risk management principles and guidelines

- **NIST RMF:** Risk Management Framework (Categorize, Select, Implement, Assess, Authorize, Monitor)
- **FAIR:** Factor Analysis of Information Risk
- **COBIT 2019:** EDM03 (Ensure risk optimization)

Security Controls Implemented

- Automated risk identification and assessment
- Risk scoring and prioritization
- Risk register and tracking
- Risk treatment and mitigation workflows
- Risk reporting and dashboards

Audit Evidence

- Risk assessment reports and methodologies
- Risk register with treatment plans
- Risk trend analysis and metrics
- Risk acceptance and mitigation records

Regulatory Alignment

- **SOX:** Section 404 (Risk assessment for financial controls)
- **GDPR:** Article 32 (Risk-based security measures), Article 35 (DPIA)
- **HIPAA:** § 164.308(a)(1)(ii)(A) (Risk analysis)
- **SOC 2:** CC3.1 (Risk assessment), CC9.1 (Risk mitigation)