CVE 2024 for the Data Lakehouse Project

Overview of my Project:

The project constructs a complete Medallion Architecture (Bronze to Silve to SQL Analysis) for 2024 CVE that is Common Vulnerabilities and Exposures dataset using Databricks.

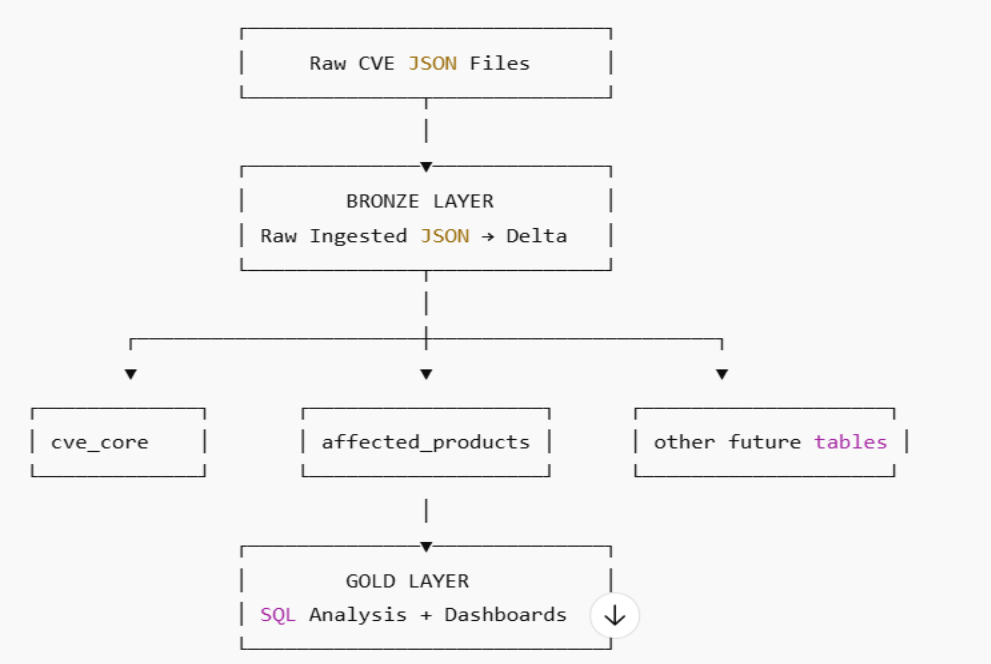
The workflow depicts the flow of raw JSON CVE records that are ingested, cleaned, normalized and analyzed to showcase the actionable cybersecurity insights.

The aim is to demonstrate:

1. Ingesting of the raw unstructured JSON data in the file
2. Normalizing deeply nested cybersecurity record
3. Building structured Delta Lake tables (Database)
4. Performing analytical queries for security insights

This documentation explains the methodology, transformations, SQL analysis, and final insights resulting out of this project.

Medallion Architecture Diagram:



DATA FLOW FOR THE PROJECT:

JSON Files → Bronze Notebook → Bronze Table → Silver Notebook → Silver Tables → SQL Analysis

2. Architecture Overview

This project makes use of the Databricks **Medallion Architecture**, which keeps data into structured quality layers that is:

Bronze layer that consists of the Raw and Ingested Data

Silver layer which has Cleaned and Normalized Tables

Gold layer having the SQL Analytics & Insights

Each layer is created and validated using the scripts and notebooks included in the project work.

3. Bronze Layer that has Raw Ingestion

File: 01_bronze_layer_2024_starter.ipynb
Objective: Load *all* 2024 CVE JSON files and build a raw Delta table.

Key insights:

- Reading the JSON using recursive file lookup
- Parsing cveMetadata fields:
    - CVE ID
    - Published Date
    - Last Modified Date
- Adding _source_file lineage tracking
- Filtering for records where year(datePublished) = 2024
- Creation of the Bronze table:
    - **Delta Path:** /Volumes/workspace/default/assignment1/bronze
    - **Table:** workspace.default.cve_bronze_records

Bronze Data Quality Checks

- Entire number of the rows is greater than 30,000
- cve_id not null
- cve_id unique across dataset

Result:

Bronze layer successfully stores the **raw but structured CVE JSON records** for the year 2024.

4. Silver Layer which has Normalized Data

File: 02_bronze_to_silver.ipynb
Objective: To Transform the raw Bronze JSON file to formatted **relational tables -Database**.

4.1 Tables Created in Silver Layer

A. cve_core (Main CVE Table)

Contains one row per CVE.

Column	Description
cve_id	Primary identifier
published_date	First disclosure date
last_modified_date	Most recent revision
cvss_score	Unified CVSS (v3.1 → v3.0 → v2 fallback)
cvss_vector	The original CVSS vector string
description	English CVE description

B. cve_affected_products

Explodes nested vendor/product/version lists.

Column	Description
cve_id	Foreign key to core table
vendor	Vendor name
product	Product name
version	Version string

Silver Transformations

- Parse nested arrays from containers.cna.affected
- Extract **English** descriptions
- Resolve multiple CVSS types:
    - Prefer CVSS v3.1
    - Then v3.0
    - Else fallback to v2
- Exploded vendor-product-version combinations
- Save tables:
    - /silver/core
    - /silver/affected_products

Silver Data Quality Checks

- Row count validation
- cve_id non-null in both tables
- Foreign key integrity:
    - Each product row should definitely map to a row in cve_core
- No malformed or missing vendor fields

Result:

Silver layer produces **clean, analytics-ready cybersecurity tables**.

ER DIAGRAM OF SILVER TABLE:

cve_core (1) ——— (∞) cve_affected_products

5. SQL Analysis Layer for the Gold Layer

File: 03_exploratory_analysis.sql

03_exploratory_analysis

Aim: Generate actionable insights from the structured Delta tables.

This SQL script covers six major analytical areas.

5.1 A. Row Count Verification

Ensures data flow is correct across layers:

- cve_bronze_records
- cve_core
- cve_affected_products

Useful for confirming:

- No data loss
- No unexpected inflation
- Proper join behavior

5.2 B. Temporal Analysis

Evaluate trends in the year 2024 CVEs:

Monthly CVE Volume

Depicts the peaks and dips in reporting trends.

Weekly CVE Distribution

Spots the high-activity disclosure weeks.

Time Lag Analysis

Computes:

- Median update time
- 90th percentile update time

Helps understand vendor patch responsiveness.

5.3 C. Severity / Risk Analysis

Derived via CVSS scores:

- Divisions of each CVE as **Critical, High, Medium, Low**
- Computes % distribution of severity
- Counts of "unscored" vulnerabilities

This quantifies the **overall risk posture** of 2024 CVEs.

5.4 D. Vendor Intelligence

Vendor-level cybersecurity analytics:

Top 25 Vendors by CVE Count

Quickly shows which vendors contributed the most vulnerabilities.

Severity Profile by Vendor

Breakdown of Critical → Low CVEs for each vendor.

Vendor Risk Summary

For each vendor:

- Total CVEs
- Average CVSS score
- Number of High + Critical CVEs

This highlights **high-risk vendors**.

5.5 E. High-Risk CVE Identification

Helping with prioritizing the remediation:

Top 50 CVEs by CVSS Score that ensures to :

Shows the most dangerous vulnerabilities.

Top Products for a Given Vendor for example: Microsoft

Vendor-specific risk profiling:

- Which Microsoft products have the most High/Critical CVEs?

5.6 F. Reusable Gold Views

Two reusable data assets were created:

cve_severity_view

- Each CVE labeled by severity

vendor_risk_summary

- Aggregated vendor-level risk

Also includes:

- % of global CVEs contributed by top 10 vendors
  Which highlights concentration of risk.

6. Final Repository Structure

/

├── 01_bronze_layer_2024_starter.ipynb # Raw data ingestion (Bronze)

├── 02_bronze_to_silver.ipynb # Data normalization (Silver)

├── 03_exploratory_analysis.sql # SQL-based Gold layer (Insights)

└── README.md # High-level summary

└── Project_Documentation.pdf (this doc if exported)

7. What This Project Demonstrates

Data Engineering Skills

- Databricks Volume ingestion
- Bronze–Silver normalization patterns
- Delta Lake table creation

Data Modeling Skills

- Flattening nested JSON
- Normalizing cybersecurity schemas
- Designing fact/dimension-like structures

Cybersecurity Analytics

- CVSS-based risk scoring
- Vendor vulnerability profiling
- Temporal vulnerability patterns

SQL Proficiency

- Window functions
- CTEs
- Aggregations, ranking, risk summaries

8. Key Learning Outcomes

By completing this project, you show mastery in:

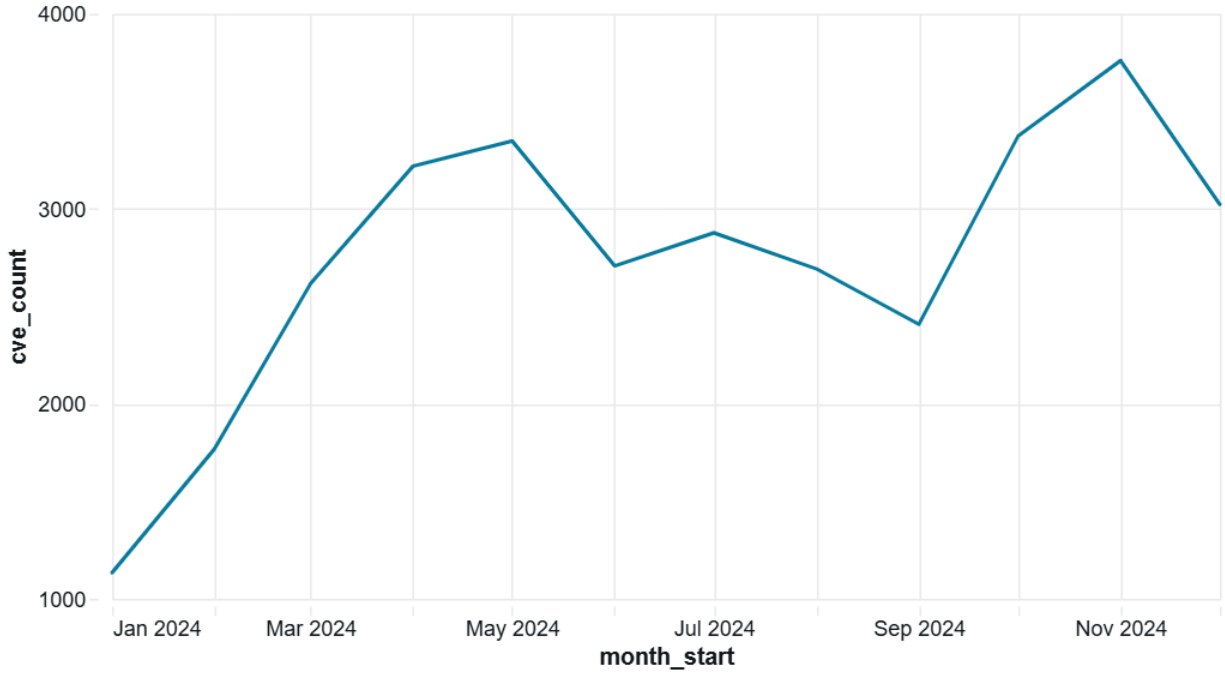
- Building a full **Medallion Architecture**
- Handling large-scale **JSON cybersecurity datasets**
- Implementing **Delta Lake best practices**
- Designing **normalized analytical tables**
- Applying SQL for **risk and vendor intelligence**
- Producing **enterprise-grade security insights**

9. Conclusion

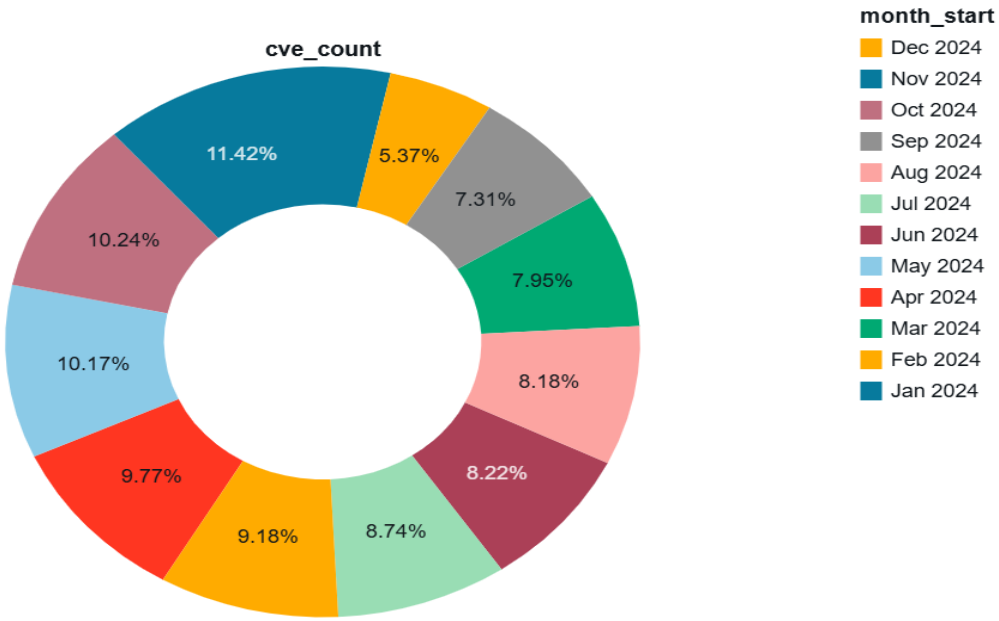
This project reflects a complete end-to-end modern data engineering pipeline for cybersecurity analysis.
From raw ingestion → normalization → analytics, all layers follow industry standards and demonstrate your proficiency in Databricks and Delta Lake.

10. VISUALIZATION:

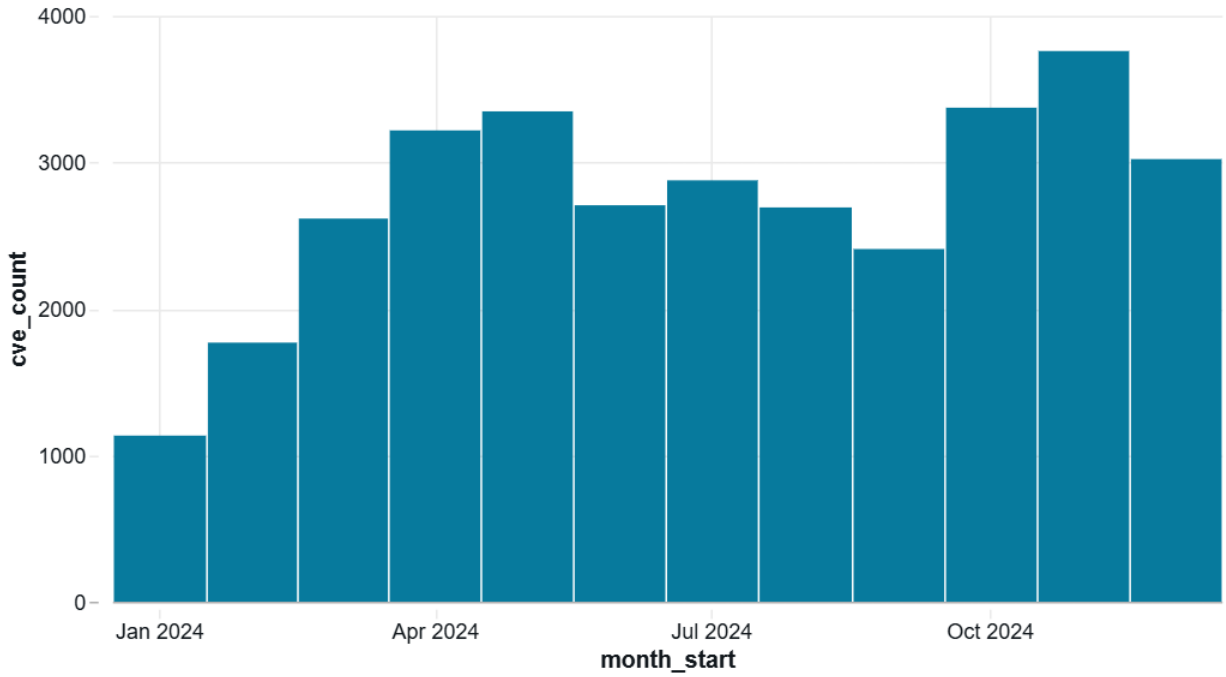
1. Monthly CVE Trend – 2024 (Line Chart)



2. Severity Distribution (Pie Chart)



3. Top 10 Vendors by Number of CVEs (Horizontal Bar)



4. Vendor Severity Heatmap (Vendor × Severity)