**MAIN INFORMATION ABOUT “TRANSFORM SCRPTS”**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Scrips** | **Main function** | **Goals** | **Characteristics** | **Errors** | **Things to imporve** |
| **duplicates eurorepoc x icsstrive x tisafe** | Data cleaning, Duplicate identification, Database unification, Data analysis, and Report generation | Unify, clean, and analyze cybersecurity incident data from multiple databases, identifying duplicates, generating relevant statistics by time, location, and industry, and presenting the results through visualizations and interactive reports. | **Modularity Each** section of the code focuses on a specific task (cleaning, analysis, visualization) **Flexibility Supports** multiple databases with varying structures **Integrated Visualizations Uses** graphs and charts to simplify data interpretation **Automation Integrates** with Datapane to generate HTML-format reports | TRUE | The references column does not seem to be handled properly when merging duplicate incidents  Ensure all charts are accessible and readable in the generated report |
| **Hackmageddon\_unification** | Consolidate data on cybersecurity incidents collected from multiple files and unify it into a single structured CSV file. | Facilitate the analysis and management of historical data related to cyberattacks by integrating information from different sources, years, and formats into a single location. | Reading files Selection of relevant columns Unification Exportation Automation | FALSE | 1. File verification 2. Error handling 3. Data standardization 4. Resource optimization 5. Result auditing |
| **join\_all\_data** | The primary purpose of the code is to unify data on cybersecurity incidents from multiple sources into a single database. These sources include various tables and files in formats such as Excel and CSV, which contain reported cybersecurity incidents. The final output is a unified CSV file and a table in a data lake. | IIntegrate and normalize data from various sources (e.g., ICSSTRIVE, TISAFE, KONBRIEFING, JAMCYBER, etc.). Ensure consistency in columns and formats, such as dates, countries, industries, and attack sources. Detect and recode inconsistencies in categorical data such as countries and industries. Save unified data in a structured format accessible for analysis and storage. | Multi-Source Integration: Unite data from multiple cyber incident sources into a single DataFrame. Preprocessing: Data cleaning and transformation (dates, lists, NaN values). Recoding: Standardization of countries and industries using predefined dictionaries. Unification: Create a common database with standardized columns such as description, date, industry, target\_country. Export: Saves the final data to a CSV file and a Delta table in the data lake. | FALSE | Modularization: Create reusable functions for repetitive tasks such as recoding or list management. Documentation: Add explanatory comments in the code. Optimization with Spark: Replace operations in Pandas with Spark for greater efficiency and scalability. Validation Automation: Incorporate tests and validations to detect inconsistencies before exporting. More Structured Pipeline: Implement logging and error tracking during execution. Early Normalization: Standardize input formats before processing each source |
| **join\_all\_together** | Process, clean, and unify data from various sources about cyber incidents. Then, combine this data into a unified structure for analysis and storage in a Data Lake table. | Unify heterogeneous data from multiple sources about cybersecurity incidents, standardize it into a common format, and store it to facilitate further analysis. This includes converting raw data from various formats into homogeneous structures and performing queries to identify trends and patterns in cyberattacks. | **1. Processing Multiple Databases:** handles various sources such as TISAFE, ICSSTRIVE, KONBRIEFING, and others. **2. Data Unification:** converts data with different formats and columns into a standardized schema. **3. Data Analysis:** enables temporal analysis and incident source breakdowns. **4. Visualization:** generates charts to identify trends, such as the number of incidents reported annually. **5. Specific Preprocessing:** includes data cleaning, date transformation, and processing of complex fields. | TRUE | **Documentation**: add more explanatory comments to improve code readability and understanding. **Data Validation:** implement strict rules to prevent errors in complex columns. **Reusable Functions:** modularize the code to avoid duplication in data transformations. **Exception Handling:** use try-except blocks to ensure continuity in case of errors. |

*Table of the main information collected about "Transform Scripts"*

## **Brief Descriptions of Scripts**

### **duplicates\_eurorepoc x icsstrive x tisafe**

This script focuses on cleaning, unifying, and analyzing cybersecurity incident data from multiple databases. It identifies duplicates, generates visualizations, and creates interactive HTML reports. While highly modular and flexible, with automated reporting, improvements are needed in handling certain columns (e.g., "references") and ensuring chart accessibility.

### **Hackmageddon\_unification**

Designed to consolidate cybersecurity incident data from various files into a structured CSV, this script simplifies historical analysis by integrating multiple sources and formats. Key processes include column selection, unification, and automation. Future enhancements could address file verification, error handling, and resource optimization.

### **join\_all\_data**

This script unifies and normalizes cybersecurity data from diverse sources like ICSSTRIVE and KONBRIEFING into a standardized database. It handles cleaning, recoding categorical data, and ensuring consistency in formats before exporting to a CSV and data lake. Improvements focus on scalability, modularization, and automated validations.

### **join\_all\_together**

This script processes and unifies heterogeneous cybersecurity data from multiple sources into a common structure for analysis and storage in a data lake. It includes data cleaning, analysis, and visualization to identify trends in incidents. Suggestions for improvement include better documentation, stricter validations, and enhanced modularity for code reuse.