Documentation for the MISP and Cypher2Sql Program

Background:

- Cypher Query Language is a graph query language
  - Allows users to store and retrieve data from a graph
  - Involves the use of the following
    - Abstraction levels
    - Abstraction nodes
    - Implementation levels
  - We can think of it as a graphical version of an SQL
- MISP
  - Database used for threat intelligence and for malware trackage
  - Displays threat Indicators
  - Accessed by the following
    - Terminal
    - Google Chrome (HTML Link)
    - Eclipse (IDE)

Requirements

- Virtual Machine (Ubuntu)
- Eclipse (Java)
  - JDK Version 8 must be used to implement and test the cypher2sql library
  - MYSQL Connector Version 5 is used to access the MISP Database through JDBC
- DBeaver (Easy access to table names and columns)
- Terminal

How to setup the Cypher2SQL Library

- Download the library as a .zip from the Cytosm Github webpage
- Extract and Import as a maven project
- Go into the POM.XL file for all the maven projects to ensure that there are not missing dependencies
- Create your gtop file and make sure to put it into the cypher2sql project (not inside any source folders)
- Create a testing source folder and add a Java class to test the library with your gtop file

My Process for Creating the GTOP file

- Due to the lack of documentation on the cypher2sql query, it was challenging to come up with the gtop file and make it as accurate to match the MISP Database (complex database with lots of tables and relations)
- Gtop is a graph topology file that maps the abstraction nodes and edges and runs on a RDBMS
- First, I created a document which consisted of the tables for events and attributes and their columns and the relationship of keys (2 main tables)

- The Gtop is broke down into 3 different layers
    - Version
        - The version I used is 1.0
    - Abstraction level
        - Abstraction Nodes
            - This consisted of the events and attributes as **types** and their **attributes** were the columns of the 2 tables
        - Abstraction Edges
            - This consisted of the properties they had common which were the IDs
    - Implementation Level
        - Graph metadata
            - Relational
        - Implementation nodes
            - This defines what the table is and what their columns consist of (column name, data type and abstraction name)
            - The 2 **types** are events and attributes, and their **attributes** are the columns of the tables
        - Implementation edges
            - Similar to abstraction edges, this is the properties they share in common which is the IDs
            - In my case, I was able to join the 2 Ids which were from the attributes and events and display them to the events (source, join and destination)
- Next Steps
    - Add more types which consist of the different MISP tables
    - Find the relationship between more tables and add them into the Gtop file to make it more scalable
    - Can be developed by following the format of the existing gtop file and its contents inside

Utilizing the MISP Database

- By reading more about the documentation of the MISP and downloading different implementations of the database, I was able to find out how to set it up and extract it through terminal and login into the MISP website on the web
- After accessing the website, I learned more about the different features and add in events, as well as attributes and display a diagram which shows the graphical representation of the events and its connections

Accessing the MISP Database

- Along with accessing the database through the website, I had set it up on Eclipse so that it can easily be accessed there, and the cypher code translated can easily be utilized in querying into the database
- To set it up on Eclipse, I had to install a MySQL connection Version 5 which was the **only** one to work

- After referencing this library, I created a method to set up the connection by adding the user and password for the database and necessary information to establish the connection
- Once performing this, I created 5 different methods that perform query such as the following
    - Querying all the tables of the database
    - Querying the events and their details (Using Cypher2SQL generated query)
    - Querying the attributes and their details (Using Cypher2SQL generated query)
    - Querying the different categories for the threat levels
    - Querying user information
- Lastly, I created a method to close the connection for the database

Improvements and Future Steps

- Build on of the gtop file (all the remaining methods can also be tested with the Cypher2SQL generated query)
- Create a method to generate cypher queries (or add in the cypher queries being used) to make it more organized