Importing and Indexing First Watch Malicious Domains Data Feed into MySQL

Posted on November 5, 2024

Abstract

This white paper provides a comprehensive guide for importing First Watch Malicious Domain data from a CSV file into a MySQL database and indexing it on the `domainName` column. The steps outlined here cover the creation of a database, the table structure design, and data import using MySQL. By following this approach, users can efficiently handle domain-related datasets for querying and analysis purposes.

1. Introduction

The growth of the internet and the expansion of domain names necessitate effective methods for storing, indexing, and analyzing domain-related data. In many cybersecurity and domain analysis use cases, data is typically collected in a CSV or JSON format, which needs to be imported into a database for processing. This white paper provides a step-by-step guide for importing domain data specifically from a CSV file into MySQL, creating a robust and efficient data storage solution for domain analysis.

2. Prerequisites

Before proceeding with the steps described in this paper, ensure the following:

- MySQL is installed and running on your Linux system. The commands will vary slightly on Microsoft Windows.
- You have sufficient privileges to create databases and tables.
- The CSV file containing domain data is available and accessible from the MySQL server.

3. Creating the MySQL Database

The first step involves creating a MySQL database to store the domain information:

Log in to MySQL

To begin, log in to the MySQL server using the following command:

```
mysql -u root -p
```

Enter the password for the root user when prompted.

Create the Database

Once logged in, create a new database named `domain_data_db`:

```
CREATE DATABASE domain_data_db;
```

Use the Database

Switch to the newly created database to prepare for table creation:

```
USE domain_data_db;
```

4. Designing and Creating the Table

The next step is to create a table to store the domain data from the CSV file. The table structure should reflect the fields in the CSV file to facilitate accurate data import.

Define the Table Structure

Create a table named `domains` that matches the fields from the CSV file:

```
CREATE TABLE domains (
  id INT AUTO_INCREMENT PRIMARY KEY,
  reason VARCHAR(255),
  domainName VARCHAR(255) UNIQUE,
  registrarName VARCHAR(255),
  registrarIANAID INT,
  whoisServer VARCHAR(255),
  nameServers TEXT,
  createdDateRaw DATETIME,
  updatedDateRaw DATETIME,
  expiresDateRaw DATETIME,
  createdDateParsed DATETIME,
```

```
    -- (add remaining columns as needed)
    INDEX (domainName)
);
```

5. Importing the CSV File into MySQL

After creating the table, the next step is importing the CSV data into the table.

Using the LOAD DATA Command

To import the data from the CSV file, use the `LOAD DATA INFILE` command. Ensure that the CSV file is accessible from the MySQL server and execute the following command:

```
LOAD DATA INFILE '/path/fwmd.YYYY-MM-DD.enterprise.daily.data.csv'
INTO TABLE domains
FIELDS TERMINATED BY ','
ENCLOSED BY ""
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(reason, domainName, registrarName, registrarIANAID, whoisServer, nameServers, createdDateRaw
```

Adjust the file path (`/path/) to point to the location of the CSV file on the MySQL server. The command will read the data from the CSV and insert it into the `domains` table. Also, depending on your MySQL configuration, you may need to use the “--local-infile=1” command-line option to use a local file. After this executes, you should see something similar to the following:

```
Query OK, 66272 rows affected, 65535 warnings (1.76 sec)
Records: 66272  Deleted: 0  Skipped: 0  Warnings: 361653
```

6. Verification

Once the import is complete, verify that the data has been correctly imported by running a simple query:

```
SELECT * FROM domains LIMIT 5;
```

This command will return the first five rows of the table, allowing you to verify that the data matches the expected format. For example:

```
mysql> select * from domains limit 5;
+----+--------+--------------------------+------------------------------+----------------+--------------------+------------
| id | reason | domainName               | registrarName                | registrarIANAID | whoisServer
+----+--------+--------------------------+------------------------------+----------------+--------------------+------------
|  1 | added  | trialparticipantforall.com | GoDaddy.com, LLC             |            146 | whois.godaddy.com
|  2 | added  | trkbj.com                | DYNADOT12 LLC                |           1867 | whois.dynadot12.com
|  3 | added  | truthslieswhatif.com     | GoDaddy.com, LLC             |            146 | whois.godaddy.com
|  4 | added  | tzsnmj.com               | West263 International Limited |           1915 | whois.hkdns.hk
|  5 | added  | uyehjwqzza.com           | Dominet (HK) Limited         |           3775 | grs-whois.aliyun.co
+----+--------+--------------------------+------------------------------+----------------+--------------------+------------
5 rows in set (0.00 sec)
```

To search for a specific domainName:

```
mysql> SELECT * FROM domains WHERE domainName = 'trialparticipantforall.com';
```

```
+----+--------+--------------------------+-----------------+---------------+-----------------+-------------------------
| id | reason | domainName               | registrarName   | registrarIANAID | whoisServer     | nameSer
+----+--------+--------------------------+-----------------+---------------+-----------------+-------------------------
|  1 | added  | trialparticipantforall.com | GoDaddy.com, LLC |           146 | whois.godaddy.com | NS41
+----+--------+--------------------------+-----------------+---------------+-----------------+-------------------------
1 row in set (0.00 sec)
```

To search for a keyword in the domainName field:

```
mysql> SELECT * FROM domains WHERE domainName like '%paypal%';
+-------+--------+---------------------------------+---------------------------------------+----------------+----------------
| id    | reason | domainName                      | registrarName                         | registrarIANAID | whoisS
+-------+--------+---------------------------------+---------------------------------------+----------------+----------------
|  5128 | added  | paypalpayments.com.mx           | PDR Ltd. d/b/a PublicDomainRegistry.com |
| 20769 | added  | paypalsolution.net              | Wix.Com Ltd.                          |           3817 | whois.wi
| 26406 | added  | paypal-pay.com                  | NameSilo, LLC                         |           1479 | whois.r
| 47177 | added  | kunden-paypal-aktualisierung.xyz | NameSilo, LLC                        |           1479 | w
+-------+--------+---------------------------------+---------------------------------------+----------------+----------------
4 rows in set (0.04 sec)
```

Python Example

```
# python findDomain.py trialparticipantforall.com

import mysql.connector
from mysql.connector import Error
import sys

def search_domain(domain_name):
```

```
    try:
        connection = mysql.connector.connect(
            host='localhost',
            database='domain_data_db',
            user='root',
            password='<YOUR_PASSWORD>'
        )
        if connection.is_connected():
            cursor = connection.cursor(dictionary=True)

            query = "SELECT * FROM domains WHERE domainName = %s"
            cursor.execute(query, (domain_name,))
            result = cursor.fetchall()

            if result:
                for row in result:
                    print(row)
            else:
                print("No domain found with the name:", domain_name)
    except Error as e:
        print("Error while connecting to MySQL", e)
    finally:
        if connection.is_connected():
            cursor.close()
            connection.close()
            print("MySQL connection is closed")

search_domain(sys.argv[1])
```

7. Conclusion

This white paper has outlined a method for importing and indexing domain data from a CSV file into a MySQL database. The guide provided covers the creation of a database, designing a table structure, and using the `LOAD DATA INFILE` command to facilitate the data import. With the `domainName` column indexed, subsequent queries can be executed more efficiently, improving the overall performance of the system.

Appendix

Sample Data

10 record sample plus header line:

```
reason,domainName,registrarName,registrarIANAID,whoisServer,nameServers,createdDateRaw,upd
added,trialparticipantforall.com,"GoDaddy.com, LLC",146,whois.godaddy.com,NS41.DOMAINCONTR
added,trkbj.com,DYNADOT12 LLC,1867,whois.dynadot12.com,sp1.gname-dns.com|sp2.gname-dns.
added,truthslieswhatif.com,"GoDaddy.com, LLC",146,whois.godaddy.com,NS03.DOMAINCONTROL.
added,tzsnmj.com,West263 International Limited,1915,whois.hkdns.hk,grapdns1.myhostadmin.net|gr
added,uyehjwqzza.com,Dominet (HK) Limited,3775,grs-whois.aliyun.com,VIP7.ALIDNS.COM|VIP8.A
added,vegannewsstand.com,"ENOM, INC.",48,WHOIS.ENOM.COM,NS-NOT-CONFIGURED-1.SUR
added,vhfamilyenterprises.com,Squarespace Domains II LLC,895,whois.squarespace.domains,ns-clo
added,vjty824.com,REALTIME REGISTER B.V.,839,whois.yoursrs.com,a.share-dns.com|b.share-dns
added,vqif074.com,Dnsgulf Pte. Ltd.,4149,whois.dnsgulf.com,NS1.DNS.COM|NS2.DNS.COM,2024-1
```

Mass Import

```
#!/bin/bash
```

```
# MySQL credentials
DB_HOST="localhost"
DB_USER="root"
DB_PASS="your_password"  # Replace with your MySQL root password
DB_NAME="domain_data_db"

# Directory containing CSV files
CSV_DIR="/path/csv_directory"  # Replace with the directory containing your CSV files

# Loop through all fwmd*.csv files in the directory
for csv_file in "$CSV_DIR"/fwmd*.csv; do
    if [[ -f "$csv_file" ]]; then
        echo "Importing $csv_file into MySQL..."

        # Import the CSV file into MySQL
        mysql --local-infile=1 -h "$DB_HOST" -u "$DB_USER" -p"$DB_PASS" "$DB_NAME" -e "
        LOAD DATA LOCAL INFILE '$csv_file'
        INTO TABLE domains
        FIELDS TERMINATED BY ','
        ENCLOSED BY '"'
        LINES TERMINATED BY '\n'
        IGNORE 1 ROWS
        (reason, domainName, registrarName, registrarIANAID, whoisServer, nameServers, createdDate
        "

        echo "$csv_file imported successfully."
    else
        echo "No fwmd*.csv files found in the directory."
    fi

done
```

```
echo "All CSV files have been processed."
```

Downloading Multiple Files

To download the entire collection using wget command:

```
wget --user="<YOUR_API_KEY" --password="<YOUR_API_KEY>" -r -np -nH --cut-dirs=6 -A "fwmd*.t
```

You can adjust the -A parameter to match your objectives.

References

First Watch Specifications: https://firstwatch.whoisxmlapi.com/specifications/datafeed-files
mySQL Documentation: https://dev.mysql.com/doc/
CSV File Handling in MySQL: https://dev.mysql.com/doc/refman/8.0/en/load-data.html

Contact Information

WHOIS, Inc. Sales: sales@whoisxmlapi.com
Support: service.desk@whoisxmlapi.com
Website: www.whoisxmlapi.com