Apply filters to SQL queries

Project description

Through the use of SQL, I used filters on various queries to investigate security events, and scan departments for information needed by the security team as follows:

Retrieve after hours failed login attempts

My organization typically closes at 6PM. I discovered a security incident that involves failed login attempts that happened after business hours (after 18:00). I need to query the `log_in_attempts` table to review after hours login activity.

The following code shows how I used SQL to do this:

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = FALSE;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        2 | apatel   | 2022-05-10 | 20:27:27   | CAN     | 192.168.205.12  |       0 |
|       18 | pwashing | 2022-05-11 | 19:28:50   | US      | 192.168.66.142  |       0 |
|       20 | tshah    | 2022-05-12 | 18:56:36   | MEXICO  | 192.168.109.50  |       0 |
```

The above query selects all data from the `log_in_attempts` table and searches for data entries that failed and have a `login_time` that is greater than 18:00, which is after operational hours.

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. I want to investigate this by reviewing all login attempts that happened on this day and the day before.

I used the following SQL query:

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       0 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       0 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71  |       0 |
```

This query selects all data from the `log_in_attempts` table that falls between the dates of 2022-05-09 and 2022-05-08.

Retrieve login attempts outside of Mexico

The security team determined that suspicious activity with login attempts did not originate in Mexico. Given this information, I need to investigate login attempts that occurred outside of Mexico.

I used the swallowing filters in SQL to do this:

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
+----------+----------+------------+------------+---------+----------------+---------+
| event_id | username | login_date | login_time | country | ip_address     | success |
+----------+----------+------------+------------+---------+----------------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       0 |
|        2 | apatel   | 2022-05-10 | 20:27:27   | CAN     | 192.168.205.12  |       0 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       0 |
```

The above query searches the data table for login attempts that did not originate from Mexico using the `NOT` and `LIKE` filters. Essentially, the query filters by looking for data in the `country` column that does not contain values of `MEX` or `MEXICO`.

Retrieve employees in Marketing

My team wishes to perform security updates on specific employee machines in the Marketing department. I am responsible for getting information on these machines.

The following query in the employees table shows how I did this:

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Marketing' AND office LIKE 'East%';
+-------------+--------------+----------+------------+----------+
| employee_id | device_id    | username | department | office   |
+-------------+--------------+----------+------------+----------+
|        1000 | a320b137c219 | elarson  | Marketing  | East-170 |
|        1052 | a192b174c940 | jdarosa  | Marketing  | East-195 |
|        1075 | x573y883z772 | fbautist | Marketing  | East-267 |
```

This query uses the `AND` filter to search for data that includes both the Marketing department, and offices that include the characters `'East'` which is usually followed by numbers like

East-170, East-320, etc. Both the department value and the office value must be true for the data to be displayed.

Retrieve employees in Finance or Sales

My team now needs to do a different security update for employees in the Sales and Finance departments.

I used the following query to identify employees in these departments:

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'Sales';
+-------------+--------------+----------+------------+-------------+
| employee_id | device_id    | username | department | office      |
+-------------+--------------+----------+------------+-------------+
|        1003 | d394e816f943 | sgilmore | Finance    | South-153   |
|        1007 | h174i497j413 | wjaffrey | Finance    | North-406   |
|        1008 | i858j583k571 | abernard | Finance    | South-170   |
```

This query uses the `OR` filter to include both department values of Finance and Sales regardless of one or the other being true/false. All data that has a department value of Finance or Sales will be shown.

Retrieve all employees not in IT

Employee machines require one more update. The IT department has already received this update, so I needed to search for all employees not in IT who needed the update.

I used the following query:

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE NOT department = 'Information Technology';
+-------------+--------------+----------+-----------------+-------------+
| employee_id | device_id    | username | department      | office      |
+-------------+--------------+----------+-----------------+-------------+
|        1000 | a320b137c219 | elarson  | Marketing       | East-170    |
|        1001 | b239c825d303 | bmoreno  | Marketing       | Central-276 |
|        1002 | c116d593e558 | tshah    | Human Resources | North-434   |
```

This query uses the `NOT` filter to exclude all data entries that have a department value of ‘Information Technology’, allowing me to see all other employees that need the required update.

Summary

The tasks above required me to be familiar with SQL filters to ensure that I could find relevant data in an efficient manner. I used the `log_in_attempts` and `employees` tables. I used the `AND`, `OR`, and `NOT` operators to filter for the specific information needed for each task, and I used `LIKE` and the percentage sign (`%`) to filter for patterns.