Apply filters to SQL queries

Project description

As a security professional at a large organization, part of my job responsibilities include investigating security issues to help keep the system secure. I recently discovered some potential security issues that involve login attempts and employee machines. I will use the MariaDB shell to run SQL queries and filter for specific sets of data.

Retrieve after hours failed login attempts

I have recently discovered a potential security incident that occurred after business hours. In order to investigate, I queried the **log_in_attempts** table in the **employees** database and reviewed after hours login activity. I then used filters in SQL to create a query that identified all failed login attempts that occurred after 18:00. The time of the login attempt is found in the **login_time** column, and the **success** column contains a value of **0** when a login attempt has failed. Receiving either a **0** or **FALSE** is indicative of a failed login attempt.

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = 0;
+----------+----------+------------+------------+---------+--------------
--+----------+
| event_id | username | login_date | login_time | country | ip_address
  | success |
```

I queried the **log_in_attempts** table and identified all employees that had a failed login attempt after 18:00. The results showed 19 login attempts were made after 18:00.

Retrieve login attempts on specific dates

We noticed that a suspicious event occurred on 2022-05-09. In order to investigate, we reviewed all login attempts on the day of the event and the day before. We used filters in SQL to create a query that returned login attempts that occurred on 2022-05-09 or 2022-05-08.

```
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 |
+----------+----------+------------+------------+---------+--------------
```

I once again queried the **log_in_attempts** table and identified all events that occurred on 2022-05-08 or 2022-05-09. 75 login attempts were made over these two days.

Retrieve login attempts outside of Mexico

We determined that although there had been suspicious login activity, this activity did not originate in Mexico. We then investigated login attempts that occurred outside of Mexico, which is referred to as both “Mex” and “Mexico”.

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
```

We used “LIKE ‘MEX%’” to select for countries that may have their country written as “Mexico” or “Mex”. The “NOT” is indicating that the results will show all login attempts not made in Mexico.

Retrieve employees in Marketing

The team now performed security updates on specific machines in the Marketing department. I needed to gather more information by querying the **employees** database. I created filters in SQL that identified all employees in the Marketing department for all offices in the East building.

```
MariaDB [organization]> SELECT * FROM employees WHERE department = "market
ing" 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 |
|        1088 | k8651965m233 | rgosh    | Marketing  | East-157 |
|        1103 | NULL         | randerss | Marketing  | East-460 |
|        1156 | a184b775c707 | dellery  | Marketing  | East-417 |
|        1163 | h679i515j339 | cwilliam | Marketing  | East-216 |
+-------------+--------------+----------+------------+----------+
7 rows in set (0.018 sec)
```

Retrieve employees in Finance or Sales

Next, we needed to perform security updates on machines for employees in the Sales and Finances departments. I created filters in SQL that identified employees in the Sales and Finances departments.

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = "Finance" OR department = "Sales";
+-------------+-------------+----------+------------+-------------+
| employee_id | device_id   | username | department | office      |
```

We identified 71 employees in both the Finance or Sales departments.

Retrieve all employees not in IT

We needed to perform one last security update. Employees in the Information Technology department have already received it, but all other departments need the update. I created a query in SQL that identified all departments except Information Technology.

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department != "Information Technology";
```

The results showed that we needed to perform updates for 161 other employees.

Summary

We made many queries and created different filters in order to view information surrounding a suspicious login attempt. We were able to retrieve information on the after hours login attempts, the country that they originated from, as well as information on login attempts made on specific dates. We then ran queries to identify employees in need of a security update based on their department.