Report. Security

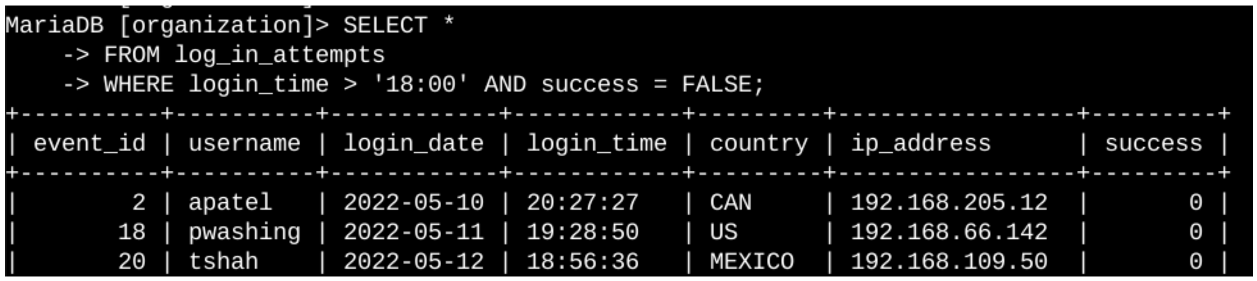
Project Description

My organization is working to make the system more secure. My job is to ensure the system is secure, investigate all potential security issues, and update employee computers as necessary. The following steps provide examples of how I used SQL with filters to perform security-related tasks.

Recover failed login attempts after hours

There was a possible security incident that occurred outside of business hours (after 6:00 p.m.). All failed after-hours login attempts should be investigated.

The following code demonstrates how I created an SQL query to filter out failed login attempts that occurred outside of business hours.



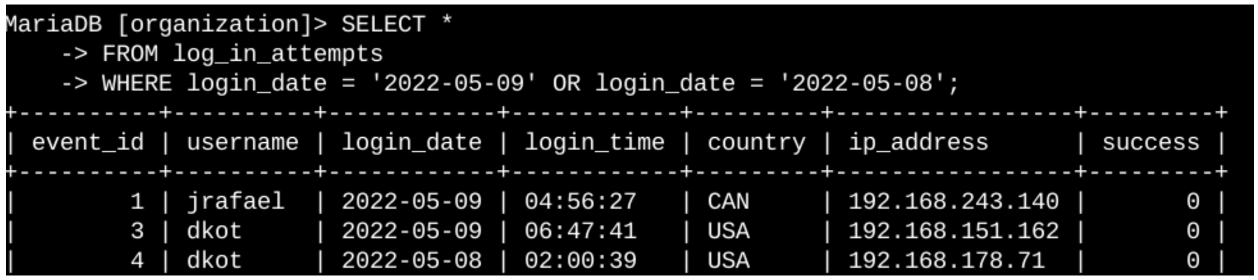
```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00' AND success = FALSE;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 first part of the screenshot is my query and the second part is a part of the result. This query filters out failed login attempts that occurred after 18:00. First, I started by selecting all the data from the `login attempts` table. Then I used a `WHERE` clause with a `AND` operator to filter my results to only output login attempts that occurred after 18:00 and were unsuccessful. The first condition is `login_time > '18:00'`, which filters out login attempts that occurred after 6:00 p.m. The second condition is `success = FALSE`, which filters out failed login attempts.

Recover login attempts on specific dates

A suspicious event occurred on 05/09/2022. Any login activity that occurred on May 9, 2022 or the day before should be investigated.

The following code demonstrates how I created a SQL query to filter login attempts that occurred on specific dates.



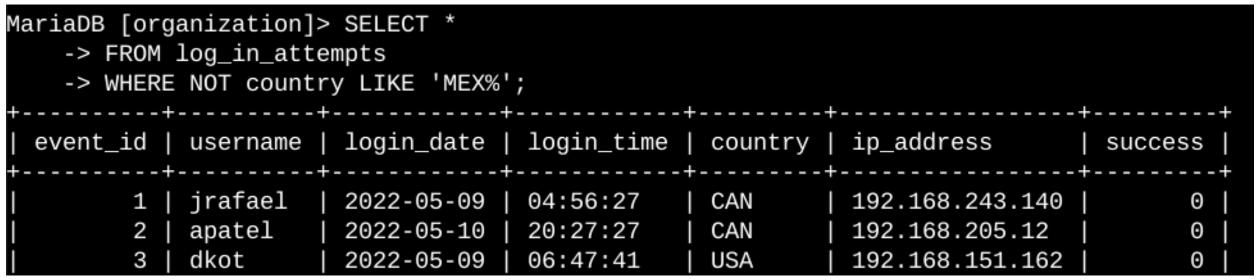
```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 |
```

The first part of the screenshot is my query and the second part is a part of the result. This query returns all login attempts that occurred on May 9, 2022 or May 8, 2022. First, I started by selecting all the data from the `login attempts` table. Then I used a `WHERE` clause with a `THE` operator to filter my results to only output login attempts that occurred on May 9, 2022 or May 8, 2022. The first condition is `login_date = '2022-05-09'`, which filters logins on 2022-05-09. The second condition is `login_date = '2022-05-08'`, which filters out logins on May 8, 2022.

Recover login attempts outside of Mexico

After researching the organization's data on login attempts, I believe there is an issue with login attempts that occurred outside of Mexico. These login attempts should be investigated.

The following code demonstrates how I created an SQL query to filter login attempts that occurred outside of Mexico.



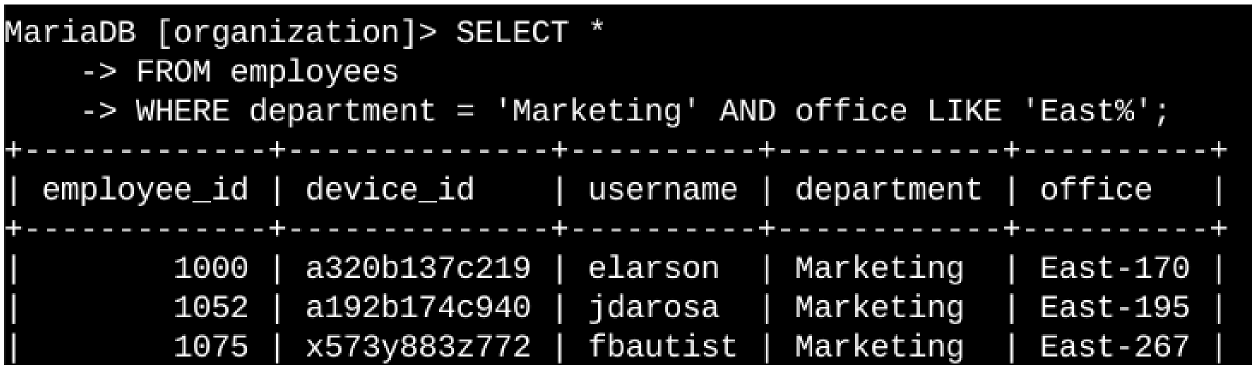
```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'MEX%';
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|        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 first part of the screenshot is my query and the second part is a part of the result. This query returns all login attempts that occurred in countries other than Mexico. First, I started by selecting all the data from the `login attempts` table. Then I used a `WHERE` clause with `NO` to filter by countries other than Mexico. I used `AS` with `MEX%` as the pattern to match because the data set represents Mexico as `MEX` and `MEXICO`. The percentage sign (`%`) represents any number of unspecified characters when used with `AS`.

Recover employees in Marketing

My team wants to upgrade the computers of certain employees in the Marketing department. To do this, I have to get information about which employee machines to update.

The following code demonstrates how I created an SQL query to filter the machines of the Marketing department employees in the East building.



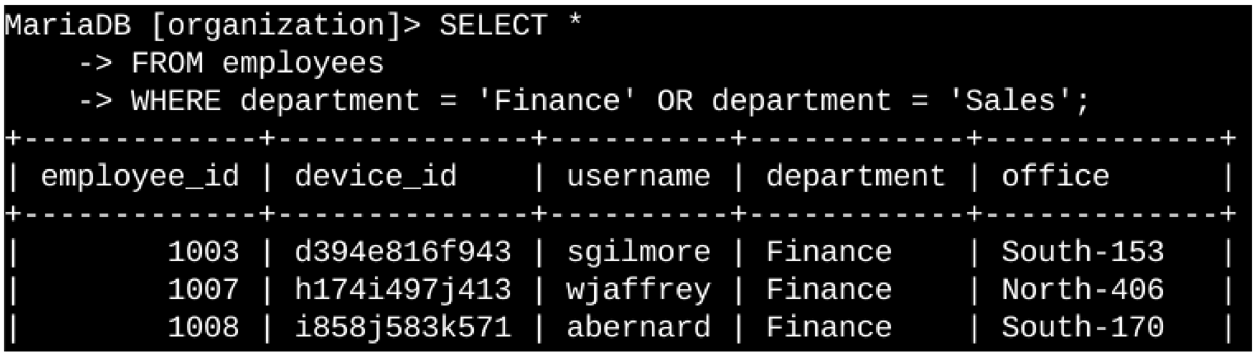
```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Marketing' AND office LIKE 'East%';
+-------------+-------------+----------+------------+----------+
| employee_id | device_id   | username | department | office   |
+-------------+-------------+----------+------------+----------+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

The first part of the screenshot is my query and the second part is a part of the result. This query returns all employees in the Marketing department in the East building. First, I started by selecting all the data from the `employees` table. Then I used a `WHERE` clause with `AND` to filter by employees who work in the Marketing department and in the East building. I used `AS` with `is%` as the pattern to match because the data in the `office` The column represents the East building with the specific office number. The first condition is the `department = 'Marketing'` portion, which filters by employees in the Marketing department. The second condition is `office LIKE 'This%'` portion, which filters for employees in the East building.

Recover employees in Finance or Sales

The machines of employees in the finance and sales departments also need to be updated. Since a different security update is needed, I only have to get information about the employees of these two departments.

The following code demonstrates how I created an SQL query to filter the machines of employees in the Finance or Sales departments.



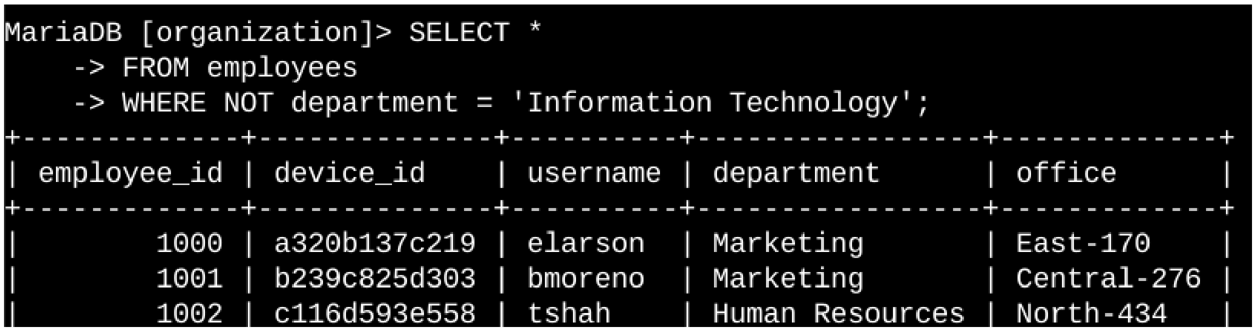
```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'Sales';
+-------------+-------------+----------+------------+-------------+
| employee_id | device_id   | username | department | office      |
+-------------+-------------+----------+------------+-------------+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

The first part of the screenshot is my query and the second part is a part of the result. This query returns all employees in the Finance and Sales departments. First, I started by selecting all the data from the `employees` table. Then I used a `WHERE` clause with `THE` to filter by employees who are in the Finance and Sales departments. I used the `THE` operator instead of `AND` because I love all the employees who are in any of the departments. The first condition is `department = 'Finance'`, which filters by employees of the Finance department. The second condition is `department = 'Sales'`, which filters by employees of the Sales department.

Bring back all non-IT employees

My team needs to perform one more security update for employees who are not in the IT department. To perform the update, I first have to obtain information about these employees.

Below is how I created a SQL query to filter machines for employees who are not in the IT department.



```
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   |
```

The first part of the screenshot is my query and the second part is a part of the result. The query returns all employees who are not in the Information Technology department. First, I started by selecting all the data from the `employees` table. Then I used a `WHERE` clause with `NO` to filter for employees who are not in this department.

Summary

Applied filters to SQL queries to obtain specific information about login attempts and employee machines. I used two different tables, `login attempts` and `employees`. I used the `AND`, `THE`, and `NO` operators to filter the specific information needed for each task. I also used `AS` and the percentage sign (`%`) wildcard to filter patterns.