Apply filters to SQL queries

Project description

A security incident occurred at my organisation after business hours. The following steps show how I have used SQL queries to investigate this incident.

Retrieve after hours failed login attempts

My team has identified that many failed attempts were made @ 18:00. So, I have utilised this information to investigate further.

The following screenshot shows the query I have created to fetch the failed login attempts after 18:00.

```
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 |
|       28 | aestrada | 2022-05-09 | 19:28:12   | MEXICO  | 192.168.27.57   |       0 |
|       34 | drosas   | 2022-05-11 | 21:02:04   | US      | 192.168.45.93   |       0 |
|       42 | cgriffin | 2022-05-09 | 23:04:05   | US      | 192.168.4.157   |       0 |
```

```
SELECT * FROM log_in_attempts
WHERE login_time > '18:00' AND Success = FALSE;
```

The SQL query above first **selects** all the fields **from** the table where logs of users' login attempts are stored and then filters them by **time** and **success**. The `WHERE` command helps to only fetch the logs where the users have tried to login after 18:00 but failed.

Retrieve login attempts on specific dates

My team has also found out that another suspicious event occurred on 2022-05-09 and they doubt that there might have been another incident the day before which requires further investigation.

The following screenshot shows the query I have created to fetch login attempts for both days (2022-05-09 AND 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 |
+----------+----------+------------+------------+---------+-----------------+---------+
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       1 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       1 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71  |       0 |
|        8 | bisles   | 2022-05-08 | 01:30:17   | US      | 192.168.119.173 |       0 |
|       12 | dkot     | 2022-05-08 | 09:11:34   | USA     | 192.168.100.158 |       1 |
|       15 | lyamamot | 2022-05-09 | 17:17:26   | USA     | 192.168.183.51  |       0 |
|       24 | arusso   | 2022-05-09 | 06:49:39   | MEXICO  | 192.168.171.192 |       1 |
|       25 | sbaelish | 2022-05-09 | 07:04:02   | US      | 192.168.33.137  |       1 |
|       26 | apatel   | 2022-05-08 | 17:27:00   | CANADA  | 192.168.123.105 |       1 |
|       28 | aestrada | 2022-05-09 | 19:28:12   | MEXICO  | 192.168.27.57   |       0 |
|       30 | yappiah  | 2022-05-09 | 03:22:22   | MEX     | 192.168.124.48  |       1 |
```

```
SELECT * FROM log_in_attempts
WHERE login_date = '2022-05-09' OR login_date = '2022-05-08';
```

The query above, first selects all the login attempts and then uses `WHERE` and `OR` operator to filter the records by the two dates.

Retrieve login attempts outside of Mexico

After investigating unusual login attempts, I have seen that most of the unusual login attempts are made outside of Mexico. So, I will be further filtering the login attempt records to investigate further.

The following screenshot shows how I created a SQL query to fetch the login attempts made 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 |       1 |
|        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 |       1 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71  |       0 |
|        5 | jrafael  | 2022-05-11 | 03:05:59   | CANADA  | 192.168.86.232  |       0 |
|        7 | eraab    | 2022-05-11 | 01:45:14   | CAN     | 192.168.170.243 |       1 |
|        8 | bisles   | 2022-05-08 | 01:30:17   | US      | 192.168.119.173 |       0 |
```

```
SELECT * FROM log_in_attempts
WHERE NOT country LIKE 'MEX%';
```

The query above, first selects all the login attempts and then uses `WHERE` , `NOT` and `LIKE` operators to fetch the login attempts **outside Mexico**. In records, I have noticed that in the country column, the country's name is represented by either the first 3 characters of name or the full name. For example: Mexico is represented by “MEX” or “MEXICO”. Therefore, I use the `'MEX%'` string to search for Mexico and I use the `NOT` operator to fetch any record that does not belong to *Mexico*.

Retrieve employees in Marketing

After the investigation my team identifies that some of the computers of some users in the **Marketing** department in the **East** block are not correctly configured and potentially may allow connection to be established after business hours. Therefore, to update the computers I have to get information on which users’ computers are misconfigured.

The following screenshot shows how I created a SQL query to fetch the Marketing employees in East block.

```
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 |
|        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 (0.001 sec)
```

```
SELECT * FROM employees
WHERE department = ‘Marketing’ AND office Like ‘East%’;
```

The query above, first selects all the employees. Then it uses the `WHERE` command to filter the employees by the ***Marketing*** department ***AND*** who work in the ***East*** office block.

Retrieve employees in Finance or Sales

My team also identifies that some of the computers' configuration of some users in the **Finance** and **Sales** departments may also require updating.

The following screenshot shows how I created a SQL query to fetch the record for employees who are in the Finance or Sales department.

```
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   |
|        1009 | NULL         | lrodriqu | Sales      | South-134   |
|        1010 | k2421212m542 | jlansky  | Finance    | South-109   |
|        1011 | l748m120n401 | drosas   | Sales      | South-292   |
|        1015 | p611q262r945 | jsoto    | Finance    | North-271   |
|        1017 | r550s824t230 | jclark   | Finance    | North-188   |
|        1018 | s310t540u653 | abellmas | Finance    | North-403   |
```

```
SELECT * FROM employees
WHERE department = 'Finance' OR department = 'Sales';
```

The query above, first selects all the employees. Then it uses the `WHERE` command to filter the employees who work in either ***Finance OR Sales*** departments.

Summary

In this task, I used SQL queries to filter out specific information on **login attempts** and **misconfigured employees' machines**. I used the `AND`, and `OR` operators alongside the `WHERE` command to filter specific information. I also use the `LIKE` to filter for patterns.