Apply filters to SQL queries

Project description

“As a security professional at a large organization, we recently discovered potential security issues that involve login attempts and employee machines. We will be examining the organization’s data from the log_in_attempts SQL tables and using SQL filters to retrieve records from different datasets, and investigate any potential security issues.”

Retrieve after hours failed login attempts

“You recently discovered a potential security incident that occurred after business hours. To investigate this, you need to query the *log_in_attempts* table and review after hours login activity. Use filters in SQL to create a query that identifies all failed login attempts that occurred after 18:00.”

```
SELECT * FROM log_in_attempts WHERE login_time > '18:00' AND success = 0;
```

Entering this query will display all login attempts after 6pm and were unsuccessful.

```
+----------+----------+------------+------------+---------+-----------------+---------+
| 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 |
|       52 | cjackson | 2022-05-10 | 22:07:07   | CAN     | 192.168.58.57   |       0 |
|       69 | wjaffrey | 2022-05-11 | 19:55:15   | USA     | 192.168.100.17  |       0 |
|       82 | abernard | 2022-05-12 | 23:38:46   | MEX     | 192.168.234.49  |       0 |
|       87 | apatel   | 2022-05-08 | 22:38:31   | CANADA  | 192.168.132.153 |       0 |
|       96 | ivelasco | 2022-05-09 | 22:36:36   | CAN     | 192.168.84.194  |       0 |
|      104 | asundara | 2022-05-11 | 18:38:07   | US      | 192.168.96.200  |       0 |
|      107 | bisles   | 2022-05-12 | 20:25:57   | USA     | 192.168.116.187 |       0 |
|      111 | aestrada | 2022-05-10 | 22:00:26   | MEXICO  | 192.168.76.27   |       0 |
|      127 | abellmas | 2022-05-09 | 21:20:51   | CANADA  | 192.168.70.122  |       0 |
|      131 | bisles   | 2022-05-09 | 20:03:55   | US      | 192.168.113.171 |       0 |
|      155 | cgriffin | 2022-05-12 | 22:18:42   | USA     | 192.168.236.176 |       0 |
|      160 | jclark   | 2022-05-10 | 20:49:00   | CANADA  | 192.168.214.49  |       0 |
|      199 | yappiah  | 2022-05-11 | 19:34:48   | MEXICO  | 192.168.44.232  |       0 |
+----------+----------+------------+------------+---------+-----------------+---------+
19 rows in set (0.002 sec)
```

We can see that there were a total of 19 failed login attempts after 6pm.

Retrieve login attempts on specific dates

“A suspicious event occurred on 2022-05-09. To investigate this event, you want to review all login attempts which occurred on this day and the day before. Use filters in SQL to create a query that identifies all login attempts that occurred on 2022-05-09 or 2022-05-08.”

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

Entering this query will display all login attempts from the dates 2022-05-09 OR 2022-05-08.

```
|       148 | daquino   | 2022-05-08 | 06:15:55   | CANADA  | 192.168.135.6   |         1 |
|       150 | nmason    | 2022-05-08 | 14:40:02   | CAN     | 192.168.204.124 |         0 |
|       151 | mabadi    | 2022-05-09 | 16:29:46   | USA     | 192.168.30.225  |         1 |
|       158 | smartell  | 2022-05-09 | 19:30:32   | MEXICO  | 192.168.190.178 |         1 |
|       161 | abellmas  | 2022-05-09 | 13:25:50   | CAN     | 192.168.180.205 |         0 |
|       162 | yappiah   | 2022-05-09 | 04:51:22   | MEXICO  | 192.168.162.100 |         0 |
|       163 | tmitchel  | 2022-05-08 | 09:21:16   | MEX     | 192.168.119.29  |         0 |
|       165 | jreckley  | 2022-05-08 | 15:28:43   | MEXICO  | 192.168.34.193  |         0 |
|       168 | jlansky   | 2022-05-08 | 13:25:42   | USA     | 192.168.210.94  |         1 |
|       169 | alevitsk  | 2022-05-08 | 08:10:43   | CANADA  | 192.168.210.228 |         0 |
|       170 | sbaelish  | 2022-05-09 | 16:43:18   | USA     | 192.168.65.113  |         0 |
|       172 | mabadi    | 2022-05-08 | 08:06:50   | US      | 192.168.180.41  |         1 |
|       178 | sgilmore  | 2022-05-08 | 12:27:22   | CAN     | 192.168.52.216  |         0 |
|       184 | alevitsk  | 2022-05-08 | 03:09:48   | CAN     | 192.168.33.70   |         0 |
|       186 | bisles    | 2022-05-09 | 04:29:17   | USA     | 192.168.40.72   |         0 |
|       187 | arusso    | 2022-05-09 | 00:36:26   | MEX     | 192.168.77.137  |         0 |
|       189 | nmason    | 2022-05-08 | 05:37:24   | CANADA  | 192.168.168.117 |         1 |
|       190 | jsoto     | 2022-05-09 | 05:09:21   | USA     | 192.168.25.60   |         0 |
|       191 | cjackson  | 2022-05-08 | 06:46:07   | CANADA  | 192.168.7.187   |         0 |
|       193 | lrodriqu  | 2022-05-08 | 07:11:29   | US      | 192.168.125.240 |         0 |
|       197 | jsoto     | 2022-05-08 | 09:05:09   | US      | 192.168.36.21   |         0 |
+-----------+-----------+------------+------------+---------+-----------------+-----------+
75 rows in set (0.001 sec)
```

We can see that there were a total of 75 login attempts between 2022-05-09 and 2022-05-08.

Retrieve login attempts outside of Mexico

“There’s been suspicious activity with login attempts, but the team has determined that this activity didn't originate in Mexico. Now, you need to investigate login attempts that occurred outside of Mexico. Use filters in SQL to create a query that identifies all login attempts that occurred outside of Mexico.”

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

By using the % operator, we can ensure that any spellings of the country Mexico, including abbreviations are included in our query.

```
|      175 | jhill    | 2022-05-10 | 00:17:09   | USA      | 192.168.130.218 |         0 |
|      177 | wjaffrey | 2022-05-11 | 00:15:55   | USA      | 192.168.144.165 |         0 |
|      178 | sgilmore | 2022-05-08 | 12:27:22   | CAN      | 192.168.52.216  |         0 |
|      179 | jclark   | 2022-05-12 | 04:08:17   | CAN      | 192.168.232.93  |         0 |
|      181 | abellmas | 2022-05-10 | 13:37:05   | CAN      | 192.168.60.111  |         0 |
|      182 | lyamamot | 2022-05-10 | 06:01:31   | USA      | 192.168.106.52  |         0 |
|      183 | nmason   | 2022-05-11 | 05:29:36   | CANADA   | 192.168.137.147 |         0 |
|      184 | alevitsk | 2022-05-08 | 03:09:48   | CAN      | 192.168.33.70   |         0 |
|      185 | jsoto    | 2022-05-10 | 13:34:58   | USA      | 192.168.151.91  |         0 |
|      186 | bisles   | 2022-05-09 | 04:29:17   | USA      | 192.168.40.72   |         0 |
|      188 | jsoto    | 2022-05-11 | 00:39:09   | USA      | 192.168.21.88   |         0 |
|      189 | nmason   | 2022-05-08 | 05:37:24   | CANADA   | 192.168.168.117 |         1 |
|      190 | jsoto    | 2022-05-09 | 05:09:21   | USA      | 192.168.25.60   |         0 |
|      191 | cjackson | 2022-05-08 | 06:46:07   | CANADA   | 192.168.7.187   |         0 |
|      192 | bisles   | 2022-05-10 | 08:32:03   | USA      | 192.168.201.40  |         1 |
|      193 | lrodriqu | 2022-05-08 | 07:11:29   | US       | 192.168.125.240 |         0 |
|      194 | jclark   | 2022-05-12 | 14:11:04   | CAN      | 192.168.197.247 |         0 |
|      195 | alevitsk | 2022-05-11 | 06:59:13   | CANADA   | 192.168.236.78  |         1 |
|      196 | acook    | 2022-05-10 | 09:56:48   | CAN      | 192.168.52.90   |         0 |
|      197 | jsoto    | 2022-05-08 | 09:05:09   | US       | 192.168.36.21   |         0 |
|      200 | jclark   | 2022-05-12 | 01:11:45   | CANADA   | 192.168.91.103  |         1 |
+----------+----------+------------+------------+----------+-----------------+-----------+
144 rows in set (0.001 sec)
```

We can see there were 144 login attempts from countries that are NOT Mexico.

Retrieve employees in Marketing

“Your team wants to perform security updates on specific employee machines in the Marketing department. You’re responsible for getting information on these employee machines and will need to query the *employees* table. Use filters in SQL to create a query that identifies all employees in the Marketing department for all offices in the East building.”

In this instance we are not provided the column names for departments and offices, we can assume that they are called ‘department’ and ‘office’, however we will do a quick check by using the select query.

```
SELECT * FROM employees;
```

Returns:

```
+-------------+--------------+----------+------------------------+-------------+
| employee_id | device_id    | username | department             | office      |
+-------------+--------------+----------+------------------------+-------------+
|        1000 | a320b137c219 | elarson  | Marketing              | East-170    |
|        1001 | b239c825d303 | bmoreno  | Marketing              | Central-276 |
```

After confirming the column names for department and office, we can now run the query:

```
SELECT * FROM employees WHERE department = 'Marketing' AND office LIKE 'East%';
```

This query will display all employees that are in the marketing department and reside in the east building offices.

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

We can see that there were a total of 7 machines in the Marketing department located in the East building that needed security updates.

Retrieve employees in Finance or Sales

“Your team now needs to perform a different security update on machines for employees in the Sales and Finance departments. Use filters in SQL to create a query that identifies all employees in the Sales or Finance departments.”

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

This query will return all employees that are either in the Sales department or Finance department.

```
|        1176 | u849v569w521 | nliu     | Sales      | West-220    |
|        1181 | z803a233b718 | sessa    | Finance    | South-207   |
|        1185 | d790e839f461 | revens   | Sales      | North-330   |
|        1186 | e281f433g404 | sacosta  | Sales      | North-460   |
|        1187 | f963g637h851 | bbode    | Finance    | East-351    |
|        1188 | g164h566i795 | noshiro  | Finance    | West-252    |
|        1195 | n516o853p957 | orainier | Finance    | East-346    |
+-------------+-------------+----------+------------+-------------+
71 rows in set (0.001 sec)
```

There are a total of 71 employees in both departments.

Retrieve all employees not in IT

“Your team needs to make one more update to employee machines. The employees who are in the Information Technology department already had this update, but employees in all other departments need it. Use filters in SQL to create a query which identifies all employees not in the IT department.”

```
SELECT * FROM employees WHERE NOT department = 'Information Technology';
```

Entering this query will display all employees who are not in the Information Technology department.

```
|        1191 | NULL          | shakimi   | Marketing        | Central-366 |
|        1194 | m340n287o441  | zwarren   | Human Resources  | West-212    |
|        1195 | n516o853p957  | orainier  | Finance          | East-346    |
|        1198 | q308r573s459  | jmartine  | Marketing        | South-117   |
|        1199 | r520s571t459  | areyes    | Human Resources  | East-100    |
+-------------+---------------+-----------+------------------+-------------+
161 rows in set (0.001 sec)
```

Giving us a total of 161 employees.

Summary

This project allowed us to explore the SQL filters of AND, OR, NOT and LIKE. Allowing us a deeper understanding of SQL and querying databases using specific filters instead of having to dig through thousands of entries manually. This allows us to be more productive and send out security updates or respond to security threats in a timely manner.