Apply filters to SQL queries

Project description:

My organization is working to make its system more secure. I am responsible for ensuring system security, investigating potential security issues, and updating employee computers as needed. Here are some examples of how I use SQL with filters to perform security-related tasks:
Identifying unauthorized users: I can use SQL to identify users who have logged into the system from unauthorized IP addresses. This helps me identify potential security breaches.
Finding vulnerable applications: I can use SQL to scan the system for vulnerable applications. This helps me identify applications that are susceptible to attacks.
Tracking suspicious activity: I can use SQL to track suspicious activity on the system. This helps me identify potential security threats.
By using SQL with filters, I help maintain my organization's system security against attacks.

Retrieve after hours failed login attempts:

In my query, I select the rows from the log_in_attempts table where the login_time is greater than 18:00 and the success column is equal to 0. This allows me to identify failed login attempts that occurred after business hours. I use the AND operator to combine both conditions in the WHERE clause. Additionally, I specify all the columns I want to retrieve in the result using the SELECT clause. This information helps me investigate potential security incidents.

```
MariaDB [organization]> clear
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_time > '18:00:00' 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 |
|       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 (0.090 sec)
```

Retrieve login attempts on specific dates:

The query selects the rows from the log_in_attempts table where the login date is equal to 2022-05-09 or 2022-05-08. This allows identifying the login attempts that occurred on those dates. The WHERE clause is used with the OR operator to filter the rows. The SELECT clause returns all columns. The query helps investigate suspicious events related to logins.

```
MariaDB [organization]>
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE login_date = '2022-05-08' OR login_date = '2022-05-09'
    -> ORDER BY login_time;
+----------+----------+------------+------------+---------+----------------+---------+
| event_id | username | login_date | login_time | country | ip_address     | success |
+----------+----------+------------+------------+---------+----------------+---------+
|      110 | mabadi   | 2022-05-09 | 00:01:54   | USA     | 192.168.90.124 |       1 |
|      117 | bsand    | 2022-05-08 | 00:19:11   | USA     | 192.168.197.187 |       0 |
|       92 | pwashing | 2022-05-08 | 00:36:12   | US      | 192.168.247.219 |       0 |
|      187 | arusso   | 2022-05-09 | 00:36:26   | MEX     | 192.168.77.137 |       0 |
|       90 | gesparza | 2022-05-09 | 00:49:05   | CANADA  | 192.168.87.201 |       0 |
|        8 | bisles   | 2022-05-08 | 01:30:17   | US      | 192.168.119.173 |       0 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71 |       0 |
|       80 | cjackson | 2022-05-08 | 02:18:10   | CANADA  | 192.168.33.140 |       1 |
|       43 | mcouliba | 2022-05-08 | 02:35:34   | CANADA  | 192.168.16.208 |       0 |
|       97 | jreckley | 2022-05-09 | 02:49:23   | MEXICO  | 192.168.32.231 |       1 |
|       32 | acook    | 2022-05-09 | 02:52:02   | CANADA  | 192.168.142.239 |       0 |
|      120 | tmitchel | 2022-05-09 | 02:58:17   | MEXICO  | 192.168.134.62 |       0 |
|      184 | alevitsk | 2022-05-08 | 03:09:48   | CAN     | 192.168.33.70  |       0 |
|       30 | yappiah  | 2022-05-09 | 03:22:22   | MEX     | 192.168.124.48 |       1 |
|      186 | bisles   | 2022-05-09 | 04:29:17   | USA     | 192.168.40.72  |       0 |
|      162 | yappiah  | 2022-05-09 | 04:51:22   | MEXICO  | 192.168.162.100 |       0 |
|        1 | jrafael  | 2022-05-09 | 04:56:27   | CAN     | 192.168.243.140 |       1 |
|       56 | acook    | 2022-05-08 | 04:56:30   | CAN     | 192.168.209.130 |       1 |
|       47 | dkot     | 2022-05-08 | 05:06:45   | US      | 192.168.233.24 |       1 |
|      190 | jsoto    | 2022-05-09 | 05:09:21   | USA     | 192.168.25.60  |       0 |
|      189 | nmason   | 2022-05-08 | 05:37:24   | CANADA  | 192.168.168.117 |       1 |
|      147 | yappiah  | 2022-05-08 | 06:04:34   | MEX     | 192.168.65.245 |       0 |
|      148 | daquino  | 2022-05-08 | 06:15:55   | CANADA  | 192.168.135.6  |       1 |
|      191 | cjackson | 2022-05-08 | 06:46:07   | CANADA  | 192.168.7.187  |       0 |
|      134 | iuduike  | 2022-05-09 | 06:46:40   | USA     | 192.168.22.115 |       1 |
|        3 | dkot     | 2022-05-09 | 06:47:41   | USA     | 192.168.151.162 |       1 |
```

Retrieve login attempts outside of Mexico:

This query will select all rows from the log_in_attempts table where the country column does not contain the string "MEX%". This allows me to identify login attempts that did not originate from Mexico. The country column stores the country of the IP address used for the login attempt. I use the WHERE clause with the NOT LIKE operator and the "MEX%" string to filter the appropriate rows. The query returns all columns. This information is helpful for investigating potential suspicious activities. To retrieve login attempts from outside of Mexico, I compare the country column with a string that does not start with "MEX".

```
MariaDB [organization]> SELECT *
    -> FROM log_in_attempts
    -> WHERE NOT country LIKE 'Mex%'
    -> ORDER BY login_date;
+----------+----------+------------+------------+---------+-----------------+---------+
| event_id | username | login_date | login_time | country | ip_address      | success |
+----------+----------+------------+------------+---------+-----------------+---------+
|       87 | apatel   | 2022-05-08 | 22:38:31   | CANADA  | 192.168.132.153 |       0 |
|      184 | alevitsk | 2022-05-08 | 03:09:48   | CAN     | 192.168.33.70   |       0 |
|       80 | cjackson | 2022-05-08 | 02:18:10   | CANADA  | 192.168.33.140  |       1 |
|       36 | asundara | 2022-05-08 | 09:00:42   | US      | 192.168.78.151  |       1 |
|      117 | bsand    | 2022-05-08 | 00:19:11   | USA     | 192.168.197.187 |       0 |
|      189 | nmason   | 2022-05-08 | 05:37:24   | CANADA  | 192.168.168.117 |       1 |
|       44 | daquino  | 2022-05-08 | 07:02:35   | CANADA  | 192.168.168.144 |       0 |
|      101 | sbaelish | 2022-05-08 | 12:01:22   | US      | 192.168.145.158 |       0 |
|      191 | cjackson | 2022-05-08 | 06:46:07   | CANADA  | 192.168.7.187   |       0 |
|       47 | dkot     | 2022-05-08 | 05:06:45   | US      | 192.168.233.24  |       1 |
|       68 | mrah     | 2022-05-08 | 17:16:13   | US      | 192.168.42.248  |       1 |
|       92 | pwashing | 2022-05-08 | 00:36:12   | US      | 192.168.247.219 |       0 |
|       83 | lrodriqu | 2022-05-08 | 08:10:23   | USA     | 192.168.67.69   |       1 |
|       53 | nmason   | 2022-05-08 | 11:51:38   | CAN     | 192.168.133.188 |       1 |
|       56 | acook    | 2022-05-08 | 04:56:30   | CAN     | 192.168.209.130 |       1 |
|      193 | lrodriqu | 2022-05-08 | 07:11:29   | US      | 192.168.125.240 |       0 |
|       72 | alevitsk | 2022-05-08 | 12:09:10   | CANADA  | 192.168.139.176 |       1 |
|       26 | apatel   | 2022-05-08 | 17:27:00   | CANADA  | 192.168.123.105 |       1 |
|       12 | dkot     | 2022-05-08 | 09:11:34   | USA     | 192.168.100.158 |       1 |
|      178 | sgilmore | 2022-05-08 | 12:27:22   | CAN     | 192.168.52.216  |       0 |
|      150 | nmason   | 2022-05-08 | 14:40:02   | CAN     | 192.168.204.124 |       0 |
|      168 | jlansky  | 2022-05-08 | 13:25:42   | USA     | 192.168.210.94  |       1 |
|      148 | daquino  | 2022-05-08 | 06:15:55   | CANADA  | 192.168.135.6   |       1 |
|      169 | alevitsk | 2022-05-08 | 08:10:43   | CANADA  | 192.168.210.228 |       0 |
|        8 | bisles   | 2022-05-08 | 01:30:17   | US      | 192.168.119.173 |       0 |
|      145 | ivelasco | 2022-05-08 | 09:06:02   | CANADA  | 192.168.39.196  |       1 |
|        4 | dkot     | 2022-05-08 | 02:00:39   | USA     | 192.168.178.71  |       0 |
```

Retrieve employees in Marketing:

I select the rows from the employees table where my department is "Marketing" and the office contains "East". This identifies my coworkers in the Marketing department in the East building. I use the WHERE clause to filter the rows based on these conditions. I compare my department to "Marketing" using the = operator and the office column to

"East" using the LIKE operator. With the SELECT clause, I retrieve all columns. This query helps me update the security of our machines.

```
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)
```

Retrieve employees in Finance or Sales:

This query selects all rows from the employees table where the department column is Sales or Finance. It will identify employees in those departments. The WHERE clause filters the rows based on the values in the department column using the OR operator. In the SELECT clause, I specify all columns to be returned. This query will allow me to identify employees in the Sales or Finance departments to perform security updates on their machines. To retrieve employees in those departments, I can use the WHERE clause and the OR operator to compare the department column to Sales or Finance.

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE department = 'Finance' OR department = 'Sales'
    -> ORDER BY employee_id;
+-------------+--------------+----------+------------+--------------+
| 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    |
|        1022 | w237x430y567 | arusso   | Finance    | West-465     |
|        1024 | y976z753a267 | iuduike  | Sales      | South-215    |
|        1025 | z381a365b233 | jhill    | Sales      | North-115    |
|        1029 | d336e475f676 | ivelasco | Finance    | East-156     |
|        1035 | j236k3031245 | bisles   | Sales      | South-171    |
|        1039 | n253o917p623 | cjackson | Sales      | East-378     |
|        1041 | p929q222r778 | cgriffin | Sales      | North-208    |
|        1044 | s429t157u159 | tbarnes  | Finance    | West-415     |
|        1045 | t567u844v434 | pwashing | Finance    | East-115     |
|        1046 | u429v921w138 | daquino  | Finance    | West-280     |
|        1047 | v109w587x644 | cward    | Finance    | West-373     |
|        1048 | w167x592y375 | tmitchel | Finance    | South-288    |
|        1049 | NULL         | jreckley | Finance    | Central-295  |
|        1050 | x132z930a114 | csimmons | Finance    | North-468    |
```

Retrieve all employees not in IT:

This query uses the WHERE clause to filter the results and display only those employees whose department is not "Information Technology". The OR operator is used to perform this comparison. The query will return all columns from the employees table.

```
MariaDB [organization]> SELECT *
    -> FROM employees
    -> WHERE NOT department LIKE 'I%T%'
    -> ORDER BY employee_id;
+-------------+-------------+----------+-----------------+-------------+
| 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   |
|        1003 | d394e816f943 | sgilmore | Finance         | South-153   |
|        1004 | e218f877g788 | eraab    | Human Resources | South-127   |
|        1005 | f551g340h864 | gesparza | Human Resources | South-366   |
|        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   |
|        1016 | q793r736s288 | sbaelish | Human Resources | North-229   |
|        1017 | r550s824t230 | jclark   | Finance         | North-188   |
|        1018 | s310t540u653 | abellmas | Finance         | North-403   |
|        1020 | u899v381w363 | arutley  | Marketing       | South-351   |
|        1022 | w237x430y567 | arusso   | Finance         | West-465    |
|        1024 | y976z753a267 | iuduike  | Sales           | South-215   |
|        1025 | z381a365b233 | jhill    | Sales           | North-115   |
|        1026 | a998b568c863 | apatel   | Human Resources | West-320    |
|        1027 | b806c503d354 | mrah     | Marketing       | West-246    |
|        1028 | c603d749e374 | aestrada | Human Resources | West-121    |
|        1029 | d336e475f676 | ivelasco | Finance         | East-156    |
|        1030 | e391f189g913 | mabadi   | Marketing       | West-375    |
|        1031 | f419g188h578 | dkot     | Marketing       | West-408    |
|        1034 | i679j565k940 | bsand    | Human Resources | East-484    |
|        1035 | j236k3031245 | bisles   | Sales           | South-171   |
|        1036 | k5501533m205 | rjensen  | Marketing       | Central-239 |
```

Summary:

In summary: I used SQL filters to obtain specific information about login attempts and employee computers. I utilized operators such as AND, OR, and NOT, as well as the LIKE operator with the percentage sign (%) wildcard to filter patterns. Examples of usage include identifying login attempts from a specific IP address, correlating login attempts with employees, and filtering by username or password. These filters enabled the retrieval of relevant data for detecting security threats and conducting investigations.