# Enhancing System Security: SQL-Based Investigations and Mitigations

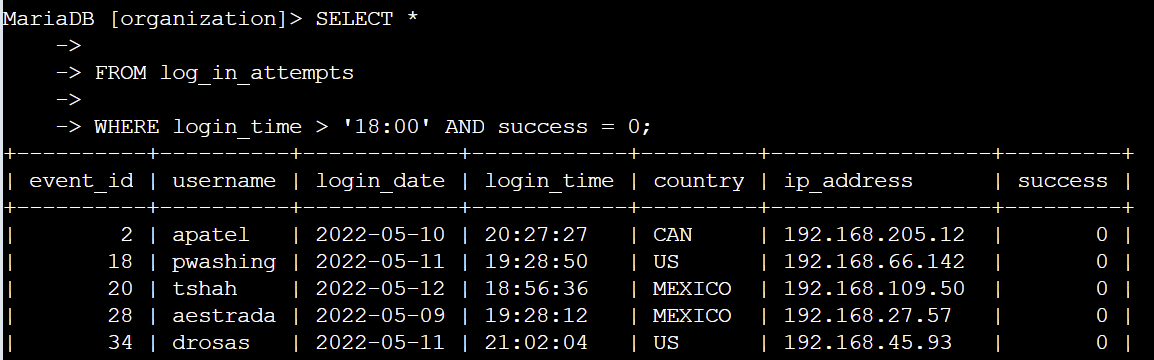
## Project description

Our organization is committed to enhancing the security of our systems. As part of my responsibilities, I ensure system safety, investigate potential security issues, and update employee computers as necessary. Below are examples of how I used SQL with filters to perform security-related tasks.

## Retrieve after hours failed login attempts

A potential security incident was detected after business hours (post 18:00). All failed login attempts during this period need to be investigated.

The following code shows how I constructed a SQL query to filter for failed login attempts occurring after business hours:

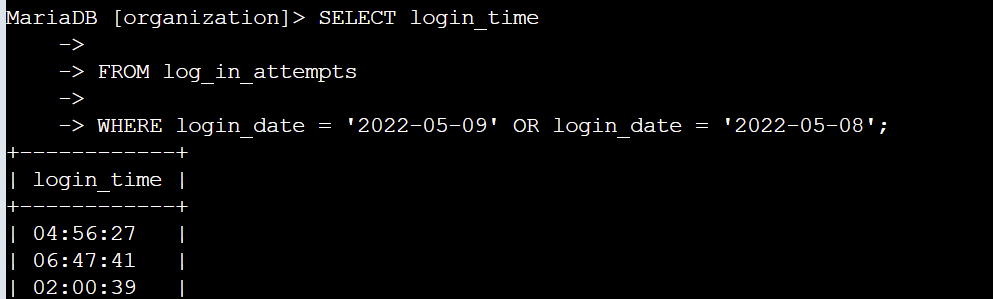


The first part of the screenshot is my query, and the second part shows a portion of the output. This query identifies failed login attempts that happened after 18:00. Initially, I selected all data from the log\_in\_attempts table. Then, I applied a WHERE clause with an AND operator to narrow down the results to login attempts that occurred after 18:00 and were unsuccessful. The first condition, login\_time > '18:00', filters for login attempts made after 18:00. The second condition, success = 0, filters for the failed login attempts.

## Retrieve login attempts on specific dates

A suspicious event was detected on 2022-05-09. Any login activity from 2022-05-09 or the previous day needs to be investigated.

The following code shows how I constructed a SQL query to filter for login attempts on specific dates:

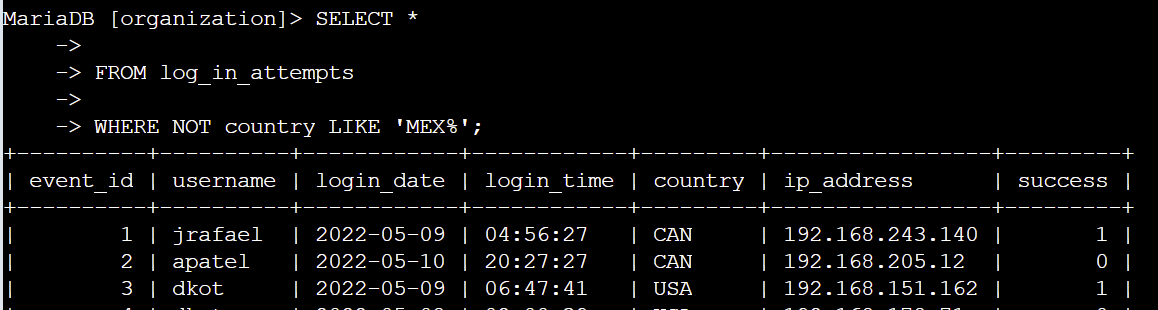


The first part of the screenshot is my query, and the second part displays a portion of the output. This query retrieves all login attempts from 2022-05-09 or 2022-05-08. Initially, I selected all data from the log\_in\_attempts table. Then, I applied a WHERE clause with an OR operator to narrow down the results to login attempts on either 2022-05-09 or 2022-05-08. The first condition, login\_date = '2022-05-09', filters for logins on 2022-05-09. The second condition, login\_date = '2022-05-08', filters for logins on 2022-05-08.

## Retrieve login attempts outside of Mexico

Upon examining the organization’s data on login attempts, I found a potential issue with those originating outside of Mexico. These login attempts require further investigation.

The following code illustrates how I created a SQL query to filter for login attempts outside of Mexico:

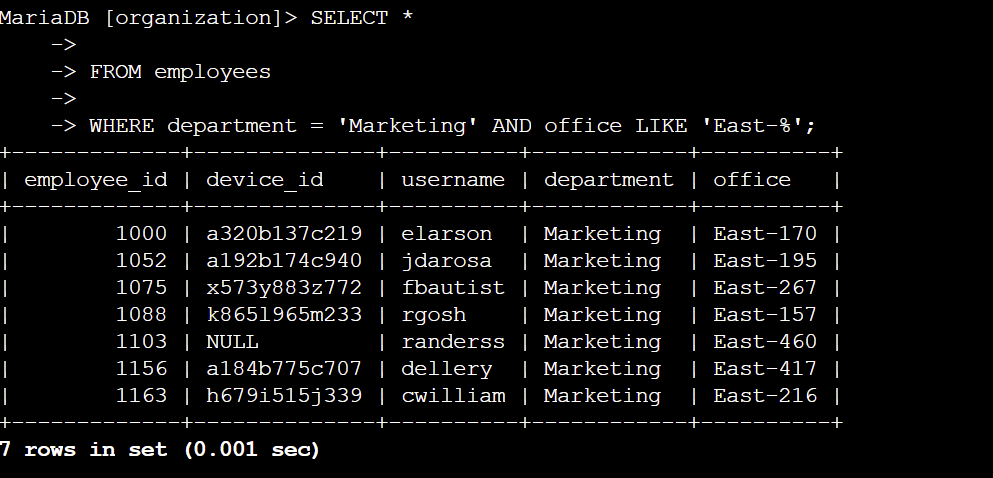


The first part of the screenshot is my query, and the second part shows a portion of the output. This query retrieves all login attempts from countries other than Mexico. Initially, I selected all data from the log\_in\_attempts table. Then, I applied a WHERE clause with NOT to exclude entries from Mexico. I used LIKE with the pattern MEX% because the dataset represents Mexico as both MEX and MEXICO. The percentage sign (%) represents any number of unspecified characters in conjunction with LIKE.

## Retrieve employees in Marketing

My team needs to update the computers for specific employees in the Marketing department. To do this, I need to identify which employee machines require updates.

The following code shows how I constructed a SQL query to filter for employee machines in the Marketing department located in the East building:

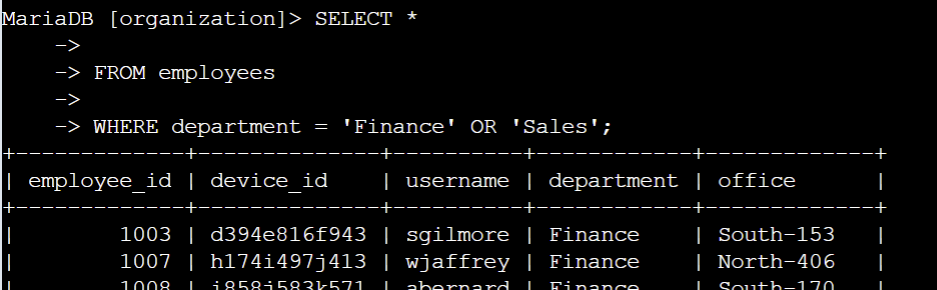


The first part of the screenshot is my query, and the second part shows a portion of the output. This query retrieves all employees in the Marketing department in the East building. Initially, I selected all data from the employees table. Then, I applied a WHERE clause with AND to filter for employees in the Marketing department and in the East building. I used LIKE with the pattern East% because the office column data represents the East building with specific office numbers. The first condition, department = 'Marketing', filters for employees in the Marketing department. The second condition, office LIKE 'East%', filters for employees in the East building.

## Retrieve employees in Finance or Sales

The machines for employees in the Finance and Sales departments also need updates. Since a different security update is required, I need to gather information on employees from these two departments specifically.

The following code shows how I created a SQL query to filter for employee machines in the Finance or Sales departments.

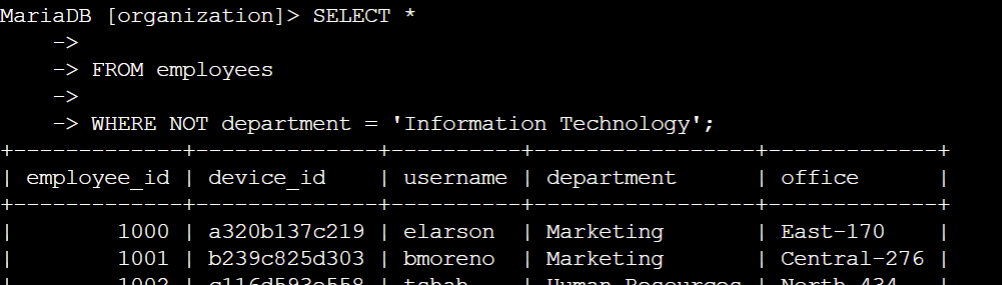


The first part of the screenshot is my query, and the second part displays a portion of the output. This query retrieves all employees in the Finance and Sales departments. Initially, I selected all data from the employees table. Then, I applied a WHERE clause with OR to filter for employees in either the Finance or Sales departments. I used the OR operator instead of AND because I want to include employees from either department. The first condition, department = 'Finance', filters for employees in the Finance department. The second condition, department = 'Sales', filters for employees in the Sales department.

## Retrieve all employees not in IT

My team needs to perform one more security update for employees who are not in the Information Technology department. To proceed with the update, I need to gather information on these employees.

The following demonstrates how I created a SQL query to filter for employee machines from employees outside the Information Technology department.



The first part of the screenshot is my query, and the second part shows a portion of the output. This query retrieves all employees not in the Information Technology department. Initially, I selected all data from the employees table. Then, I applied a WHERE clause with NOT to exclude employees from this department.

## Summary

I applied filters to SQL queries to extract specific information on login attempts and employee machines. I used the log\_in\_attempts and employees tables the AND, OR, and NOT operators to obtain the required data for each task. Additionally, I used the LIKE operator together with the percentage sign (%) wildcard to match specific patterns.