SQL Fundamentals: Device & Login Analysis

Assignment Overview

In this lab, I stepped into the role of a **security analyst** tasked with answering two critical questions every organization must ask:

1. **Which employee devices need updates?**
2. **Is there any unusual login activity that could indicate risk?**

Using the `organization` database in a MariaDB environment, I applied foundational SQL skills, `SELECT`, `FROM`, and `ORDER BY`, to retrieve, inspect, and organize data in a way that supports real-world security decisions.

This assignment mirrors common SOC and security operations workflows, where analysts must quickly pull accurate data, validate assumptions, and identify anomalies without relying on graphical tools.

Scenario Context

The organization maintains two key tables:

- **`machines`** – Tracks employee devices, operating systems, and patch history
- **`log_in_attempts`** – Records user login activity, including location, date, and time

As a security analyst, my responsibility was to:

- Identify devices that may require updates
- Review login activity for geographic and time-based anomalies
- Organize findings in a way that highlights potential risks

Task 1: Retrieving Employee Device Data

Objective

Determine which employee devices may require attention by reviewing system and patch information.

Approach

I began by retrieving **all records** from the `machines` table to understand the full scope of devices in the environment. From there, I narrowed my focus to specific attributes relevant to patching and email security.

Key Queries Executed

- Retrieved all device records using a wildcard selector
- Isolated **device IDs and email clients** to understand email exposure
- Extracted **operating system and patch dates** to assess update status

Insight

This step reflects a common security hygiene task: ensuring systems are patched and standardized. Being able to quickly isolate patch dates and operating systems is essential for vulnerability management and compliance audits.

Task 2: Investigating Login Activity

Objective

Analyze login attempts to identify potentially suspicious behavior.

Approach

I examined login activity from multiple angles:

1. **Geographic validation** – ensuring logins originated from expected regions
2. **Time-based review** – checking for access outside normal working hours
3. **Full visibility** – reviewing all login attempt details for context

Key Queries Executed

- Selected login event IDs and countries to verify geographic legitimacy
- Retrieved usernames with login dates and times to assess work-hour access
- Pulled all login attempt data for comprehensive analysis

Insight

This task reinforced how even simple queries can surface red flags. Geographic outliers or late-night access patterns are often the first indicators analysts investigate during an incident triage.

Task 3: Ordering Login Attempts

Objective

Organize login data chronologically to reveal patterns and anomalies.

Approach

Using the `ORDER BY` clause, I structured login data by:

- **Login date** to identify earliest activity
- **Login time** to pinpoint exact access sequences

Key Queries Executed

- Ordered login attempts by date
- Further refined results by ordering on both date and time

Insight

Sorting data is more than cosmetic, it enables pattern recognition. Chronological ordering helps analysts detect unusual access timing, brute-force attempts, or coordinated activity.

Key Skills Demonstrated

- Writing precise SQL `SELECT` queries
- Extracting targeted columns from large datasets
- Using `ORDER BY` for chronological analysis
- Thinking like a security analyst, not just a database user

Conclusion

This lab provided hands-on experience with **practical SQL analysis in a security context**. Rather than querying data for its own sake, each step supported a broader investigative goal, protecting systems, validating user behavior, and maintaining organizational security.