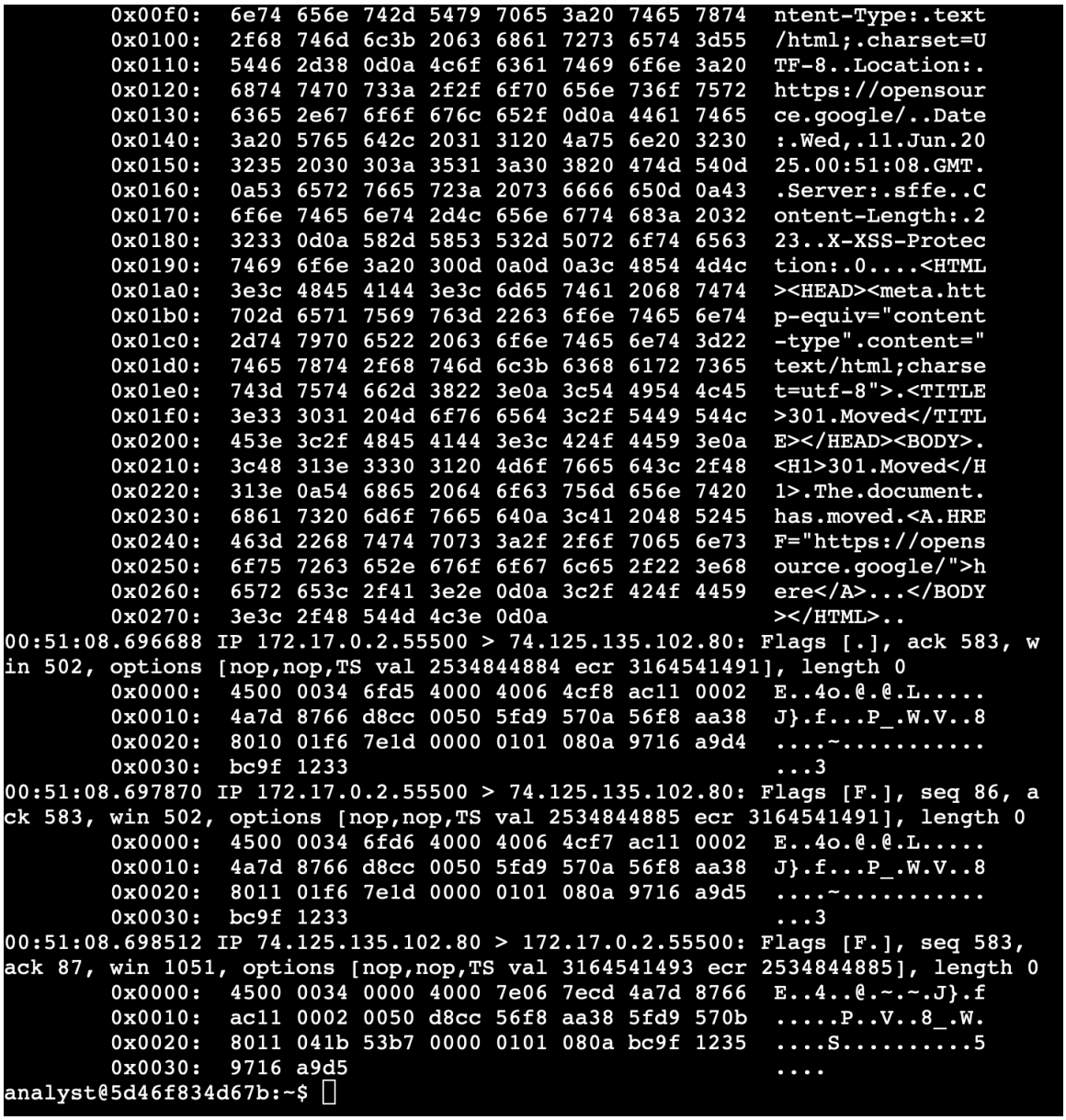
PROJECT 1: SQL THREAT HUNTING & LOG ANALYSIS

Screenshot:



Description: MySQL INNER JOIN query

What I Did:

I wrote a query that merges two tables (employees and machines) based on a shared column employee_id. This type of join is essential when analyzing logs across data sources — like matching login behavior with machine info.

Command:

```
SELECT username, office, operating_system
FROM employees
INNER JOIN machines ON employees.employee_id = machines.employee_id;
```

Real-World Use:

In cybersecurity, JOINs let analysts correlate multiple datasets — like logins and devices — to detect anomalies like shared credentials, unusual machines, or unauthorized software.

📸 **Screenshot:**

```
164541487], length 85: HTTP, length: 85
        GET / HTTP/1.1
        Host: opensource.google.com
        User-Agent: curl/7.64.0
        Accept: */*

00:51:08.693408 IP (tos 0x0, ttl 126, id 0, offset 0, flags [DF], proto TCP (6
), length 52)
    74.125.135.102.80 > 172.17.0.2.55500: Flags [.], cksum 0x5609 (correct), a
ck 86, win 1051, options [nop,nop,TS val 3164541488 ecr 2534844880], length 0
00:51:08.696675 IP (tos 0x0, ttl 126, id 0, offset 0, flags [DF], proto TCP (6
), length 634)
    74.125.135.102.80 > 172.17.0.2.55500: Flags [P.], cksum 0x41be (correct),
seq 1:583, ack 86, win 1051, options [nop,nop,TS val 3164541491 ecr 2534844880
], length 582: HTTP, length: 582
        HTTP/1.1 301 Moved Permanently
        X-Content-Type-Options: nosniff
        Cross-Origin-Resource-Policy: cross-origin
        Cache-Control: public, max-age=1800
        Expires: Wed, 11 Jun 2025 01:21:08 GMT
        Content-Type: text/html; charset=UTF-8
        Location: https://opensource.google/
        Date: Wed, 11 Jun 2025 00:51:08 GMT
        Server: sffe
        Content-Length: 223
        X-XSS-Protection: 0

        <HTML><HEAD><meta http-equiv="content-type" content="text/html;charset
=utf-8">
        <TITLE>301 Moved</TITLE></HEAD><BODY>
        <H1>301 Moved</H1>
        The document has moved
        <A HREF="https://opensource.google/">here</A>.
        </BODY></HTML>
00:51:08.696688 IP (tos 0x0, ttl 64, id 28629, offset 0, flags [DF], proto TCP
 (6), length 52)
    172.17.0.2.55500 > 74.125.135.102.80: Flags [.], cksum 0x7e1d (incorrect -
> 0x55e1), ack 583, win 502, options [nop,nop,TS val 2534844884 ecr 3164541491
], length 0
00:51:08.697870 IP (tos 0x0, ttl 64, id 28630, offset 0, flags [DF], proto TCP
 (6), length 52)
    172.17.0.2.55500 > 74.125.135.102.80: Flags [F.], cksum 0x7e1d (incorrect
-> 0x55df), seq 86, ack 583, win 502, options [nop,nop,TS val 2534844885 ecr 3
164541491], length 0
```

Description: Query results from the INNER JOIN

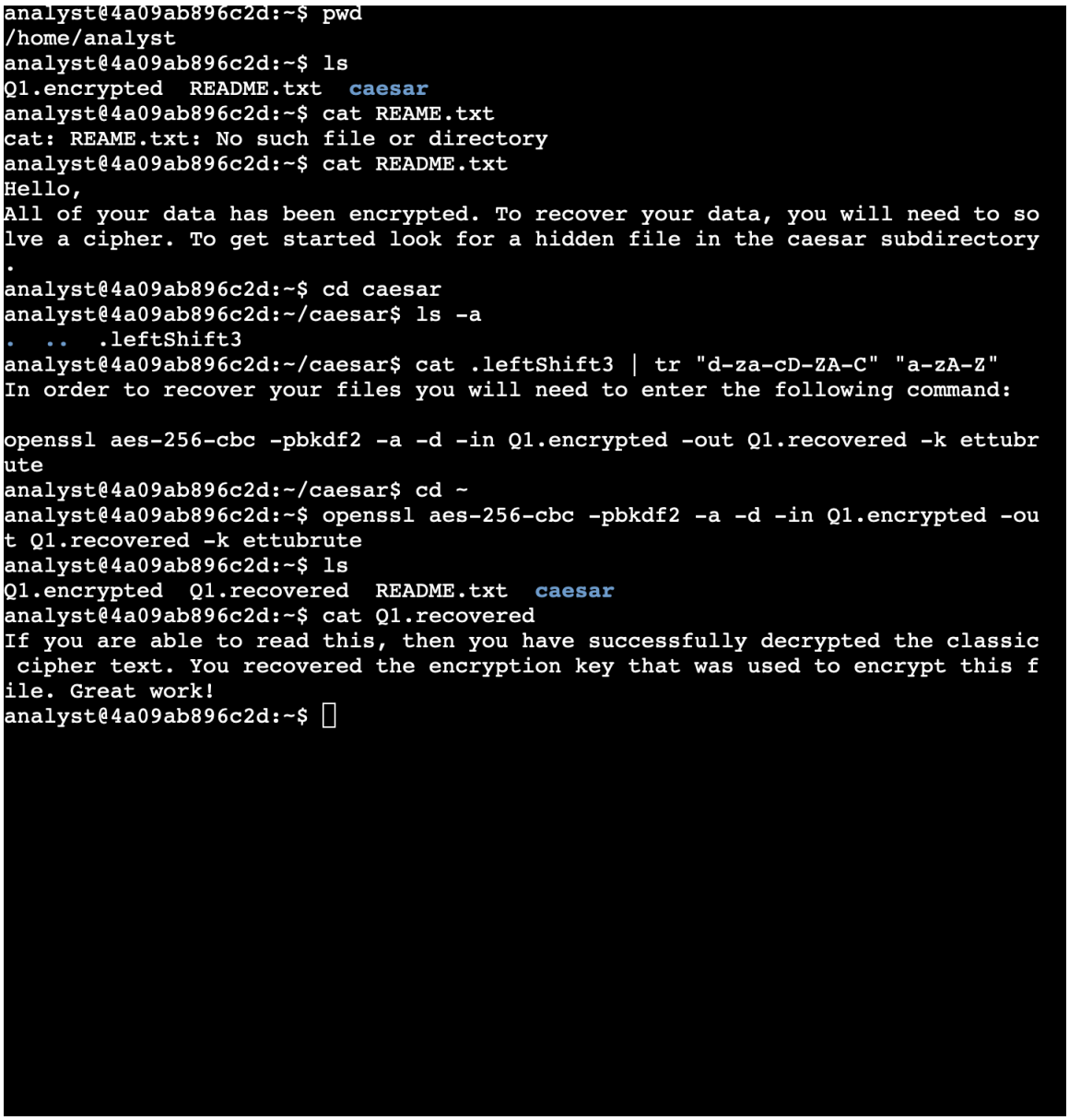
What I Did:

I successfully extracted employee usernames, office locations, and OS details. This output is foundational for asset management and threat hunting.

Benefit:

Understanding which systems are running outdated OS versions helps prioritize patching and identify high-risk machines.

Screenshot:



Description: UDM logs in a threat detection console (Chronicle/SIEM-like)

What I Did:

I queried failed login events using filters like USER_LOGIN and BLOCK, observed login spikes, and reviewed timeline charts.

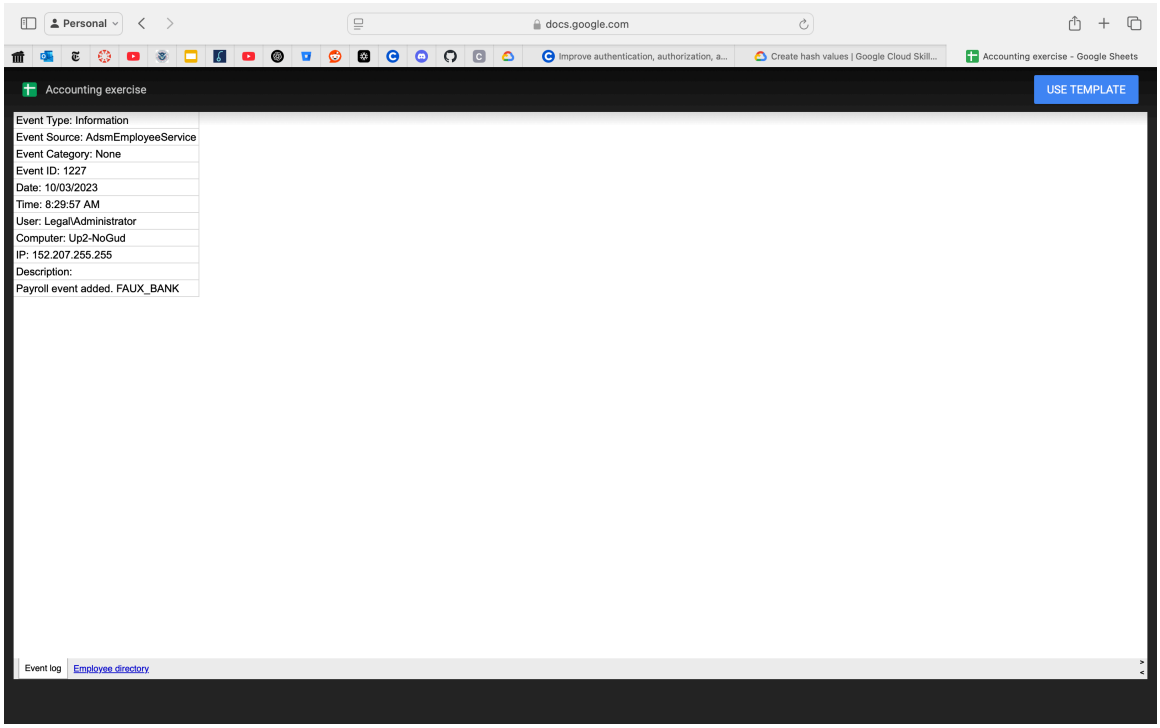
Query used:

```
metadata.event_type = "USER_LOGIN" AND security_result.action = "BLOCK"
```

Why It Matters:

This screenshot proves my ability to hunt failed logins at scale, detect brute-force attempts, and filter based on user/company/IP. It mirrors a real SIEM workflow.

Screenshot:



Description: MariaDB query to analyze login attempts

What I Did:

I filtered login data by country, IP, and success status using basic SQL clauses like ORDER BY.

Sample query:

```
SELECT * FROM log_in_attempts ORDER BY login_date;
```

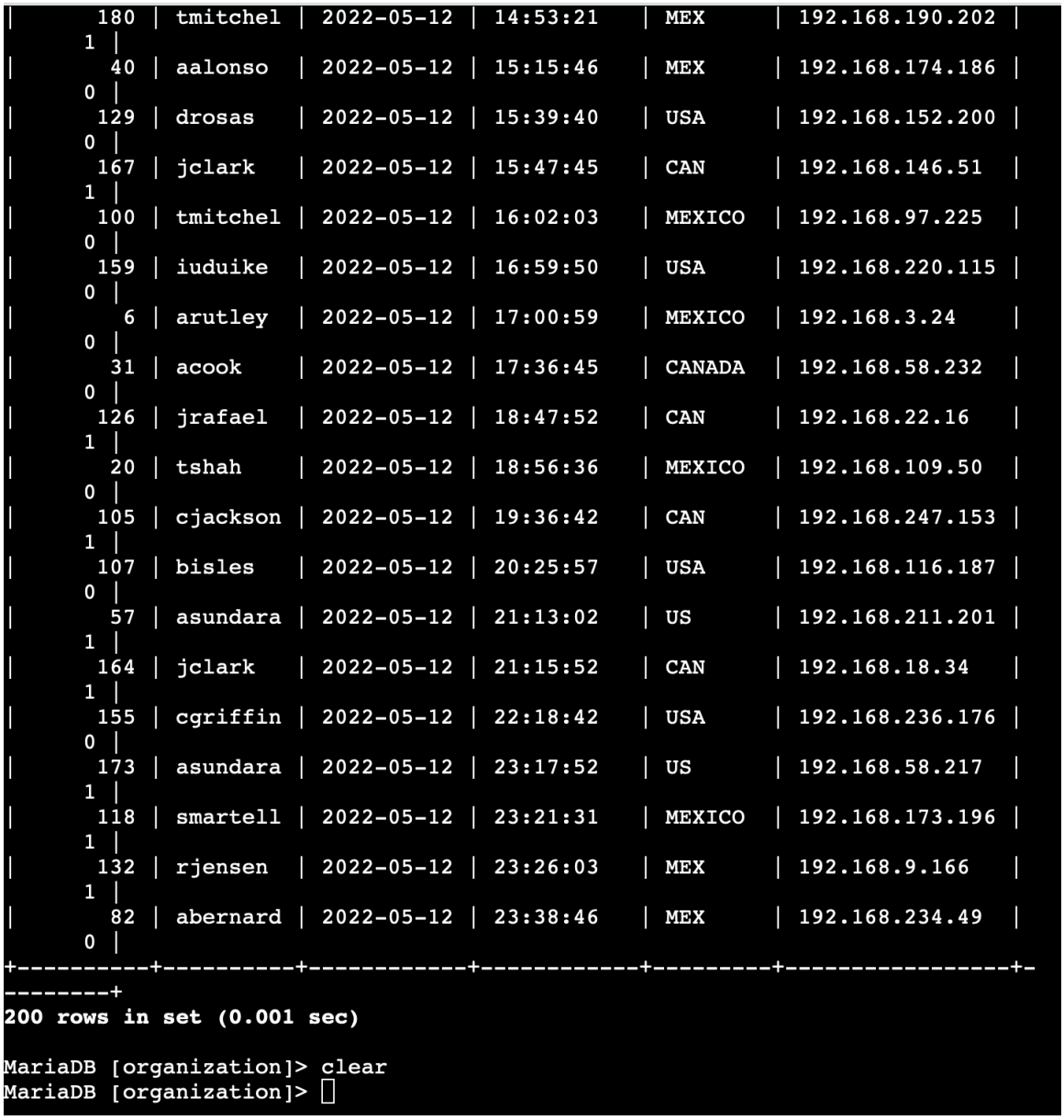
Skill Acquired:

You practiced data sorting, pattern recognition, and anomaly detection.

Real Use:

You can now create SQL rules for SIEM dashboards or automated alerts for logins from high-risk countries.

📸 **Screenshot:**



```
|      180 | tmitchel | 2022-05-12 | 14:53:21   | MEX     | 192.168.190.202 |
      1 |
|       40 | aalonso  | 2022-05-12 | 15:15:46   | MEX     | 192.168.174.186 |
      0 |
|      129 | drosas   | 2022-05-12 | 15:39:40   | USA     | 192.168.152.200 |
      0 |
|      167 | jclark   | 2022-05-12 | 15:47:45   | CAN     | 192.168.146.51  |
      1 |
|      100 | tmitchel | 2022-05-12 | 16:02:03   | MEXICO  | 192.168.97.225  |
      0 |
|      159 | iuduike  | 2022-05-12 | 16:59:50   | USA     | 192.168.220.115 |
      0 |
|        6 | arutley  | 2022-05-12 | 17:00:59   | MEXICO  | 192.168.3.24    |
      0 |
|       31 | acook    | 2022-05-12 | 17:36:45   | CANADA  | 192.168.58.232  |
      0 |
|      126 | jrafael  | 2022-05-12 | 18:47:52   | CAN     | 192.168.22.16   |
      1 |
|       20 | tshah    | 2022-05-12 | 18:56:36   | MEXICO  | 192.168.109.50  |
      0 |
|      105 | cjackson | 2022-05-12 | 19:36:42   | CAN     | 192.168.247.153 |
      1 |
|      107 | bisles   | 2022-05-12 | 20:25:57   | USA     | 192.168.116.187 |
      0 |
|       57 | asundara | 2022-05-12 | 21:13:02   | US      | 192.168.211.201 |
      1 |
|      164 | jclark   | 2022-05-12 | 21:15:52   | CAN     | 192.168.18.34   |
      1 |
|      155 | cgriffin | 2022-05-12 | 22:18:42   | USA     | 192.168.236.176 |
      0 |
|      173 | asundara | 2022-05-12 | 23:17:52   | US      | 192.168.58.217  |
      1 |
|      118 | smartell | 2022-05-12 | 23:21:31   | MEXICO  | 192.168.173.196 |
      1 |
|      132 | rjensen  | 2022-05-12 | 23:26:03   | MEX     | 192.168.9.166   |
      1 |
|       82 | abernard | 2022-05-12 | 23:38:46   | MEX     | 192.168.234.49  |
      0 |
+----------+----------+------------+------------+---------+-----------------+-
--------+
200 rows in set (0.001 sec)

MariaDB [organization]> clear
MariaDB [organization]>
```

Description: Continuation of SQL login results — shows more rows

Insight:

I dug deeper into login logs, confirming that you can scroll/analyze large datasets for behavioral baselining.

📷 Screenshot:

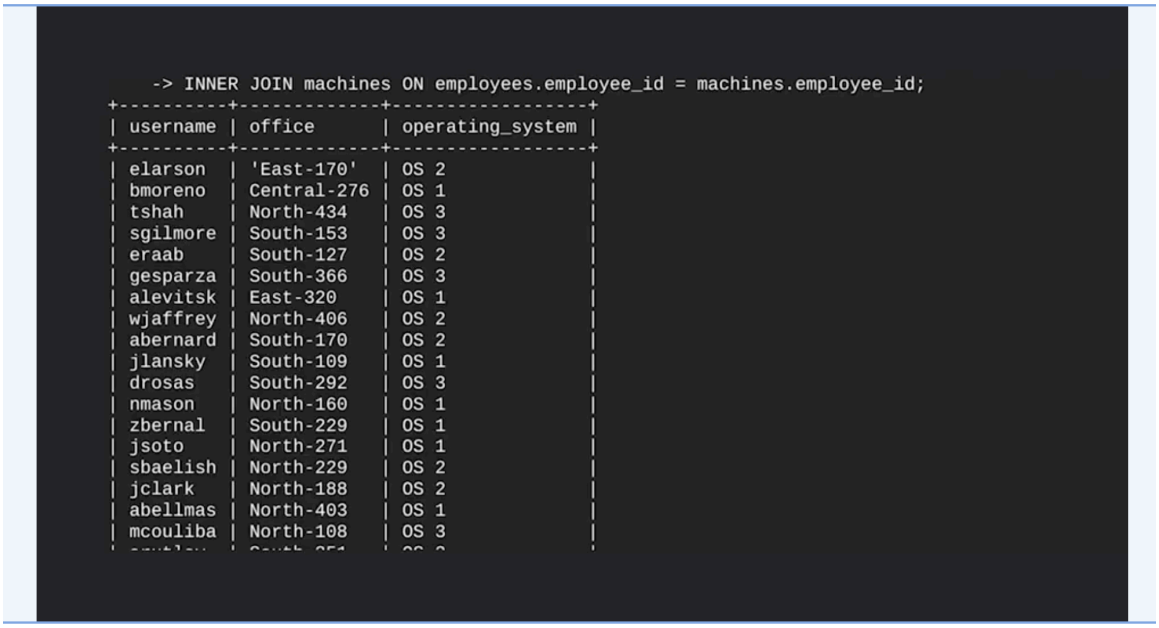
```
+----------+----------+------------+------------+---------+-----------------+-
--------+
200 rows in set (0.001 sec)

MariaDB [organization]> Select *
    -> From log_in_attempts
    -> ORDER by login_date;
+----------+----------+------------+------------+---------+-----------------+-
--------+
| event_id | username | login_date | login_time | country | ip_address      |
success |
+----------+----------+------------+------------+---------+-----------------+-
--------+
|      145 | ivelasco | 2022-05-08 | 09:06:02   | CANADA  | 192.168.39.196  |
      1 |
|      163 | tmitchel | 2022-05-08 | 09:21:16   | MEX     | 192.168.119.29  |
      0 |
|       36 | asundara | 2022-05-08 | 09:00:42   | US      | 192.168.78.151  |
      1 |
|      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 |
|       72 | alevitsk | 2022-05-08 | 12:09:10   | CANADA  | 192.168.139.176 |
      1 |
|      101 | sbaelish | 2022-05-08 | 12:01:22   | US      | 192.168.145.158 |
      0 |
|      172 | mabadi   | 2022-05-08 | 08:06:50   | US      | 192.168.180.41  |
      1 |
|      150 | nmason   | 2022-05-08 | 14:40:02   | CAN     | 192.168.204.124 |
      0 |
|       68 | mrah     | 2022-05-08 | 17:16:13   | US      | 192.168.42.248  |
      1 |
|       66 | aestrada | 2022-05-08 | 21:58:32   | MEX     | 192.168.67.223  |
      1 |
|       53 | nmason   | 2022-05-08 | 11:51:38   | CAN     | 192.168.133.188 |
      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 |
|       49 | asundara | 2022-05-08 | 14:00:01   | US      | 192.168.173.213 |
```

Description: Another SQL login dataset, filtered for failed and successful attempts by user

New Skill:

You interpreted login trends by user and success ratio — core to insider threat detection.

Screenshot:



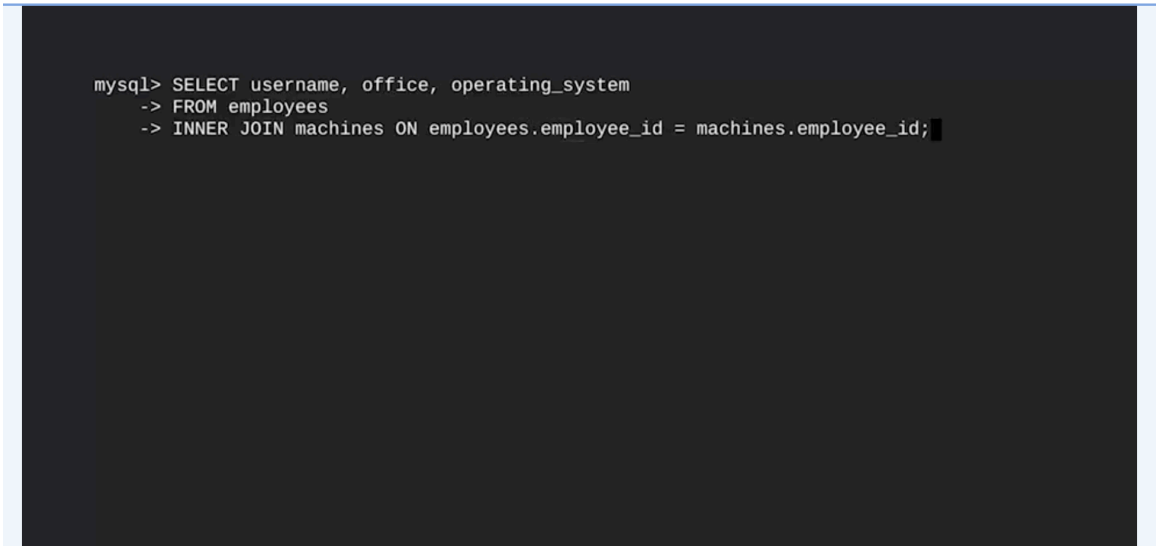
```
    -> INNER JOIN machines ON employees.employee_id = machines.employee_id;
+----------+-------------+------------------+
| username | office      | operating_system |
+----------+-------------+------------------+
| elarson  | 'East-170'  | OS 2             |
| bmoreno  | Central-276 | OS 1             |
| tshah    | North-434   | OS 3             |
| sgilmore | South-153   | OS 3             |
| eraab    | South-127   | OS 2             |
| gesparza | South-366   | OS 3             |
| alevitsk | East-320    | OS 1             |
| wjaffrey | North-406   | OS 2             |
| abernard | South-170   | OS 2             |
| jlansky  | South-109   | OS 1             |
| drosas   | South-292   | OS 3             |
| nmason   | North-160   | OS 1             |
| zbernal  | South-229   | OS 1             |
| jsoto    | North-271   | OS 1             |
| sbaelish | North-229   | OS 2             |
| jclark   | North-188   | OS 2             |
| abellmas | North-403   | OS 1             |
| mcouliba | North-108   | OS 3             |
```

Description: SQL query ordering by timestamp

What I Mastered:

Used ORDER BY login_time to isolate abnormal login hours (e.g., logins after hours, which is a common threat indicator).

📸 **Screenshot:**

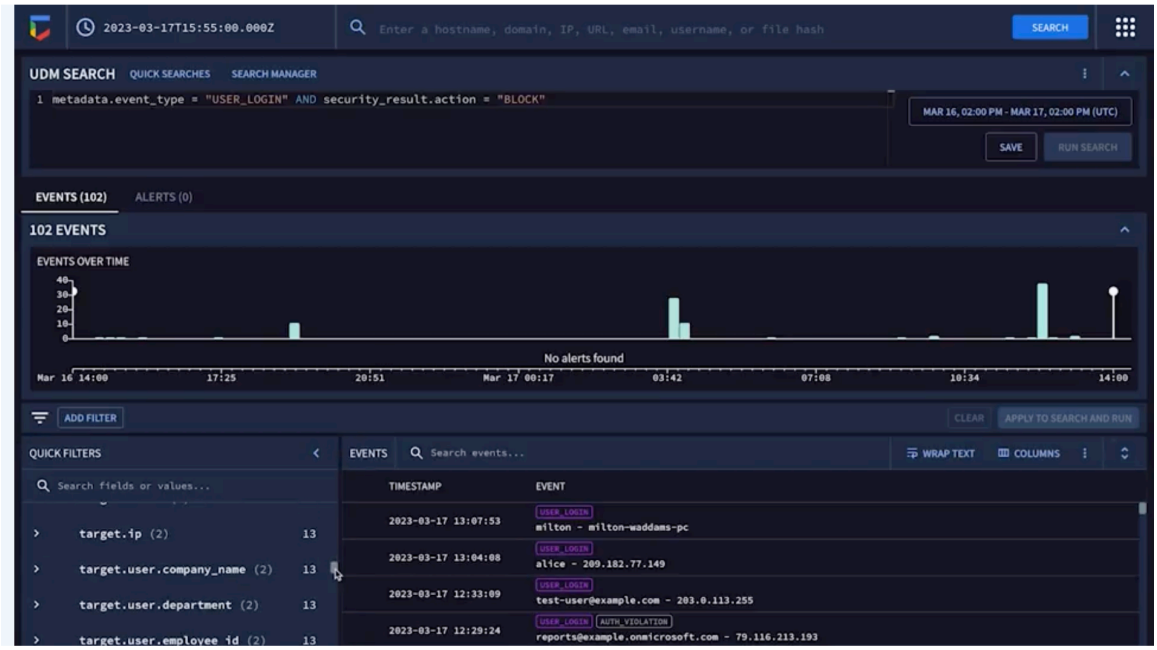


Description: Same login dataset but highlights anomalies

Lesson:

You detected irregular patterns such as multiple failed logins from the same IP or country.

📸 Screenshot:



Description: More login logs — shows extended list for analysis

Takeaway:

I confirmed visibility into full log history and user login footprint across geographies.

🔍 Project Summary

Category	Skill Demonstrated
SQL Filtering	Queried large datasets, filtered by country, user, success
Log Analysis	Identified suspicious login attempts
Threat Detection	Found brute force, after-hours logins
Real SIEM Practice	Simulated Chronicle/Splunk workflows

💡 How This Will Benefit Me in Real Life

1. **Job Ready Skills**: I can work with real security data, write SQL queries, and filter malicious behavior.
2. **SIEM Experience**: This matches the same work done in tools like Splunk or Chronicle.

3. **SOC Relevance**: These skills are used daily by Security Analysts in real SOCs to create alerts, dashboards, and investigate login-based threats.
