UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
DEHRADUN

Database Management System

AI-Enhanced Database Systems Lab Report

MTECH-COMPUTER SCIENCE ENGINEERING
CYBER SECURITY AND FORENSICS

Name: Jigesh Sheoran
SAP ID: 590025428

Topic: Cybersecurity Log Analysis & Threat Detection Using GPT-5.1

1. Introduction

Modern systems generate a ridiculous amount of security logs — authentication logs, network traffic, firewall alerts, and intrusion detection system (IDS) events.
Manually analyzing this data is slow, tiring, and honestly unrealistic at scale.

Goal:

Store cybersecurity logs in a structured database (MySQL)
Use GPT-5.1 to analyze those logs for anomalies, threats, suspicious patterns
Document how AI can enhance traditional database systems

So this report covers the entire workflow from designing the database to using GPT-5.1 as an "AI SOC Analyst".

2. Problem Statement

Security teams struggle with:

- Increasing log volume
- Noisy alerts
- Repetitive manual queries
- Difficulty spotting subtle anomalies
- Limited context behind raw log entries

This project shows how a DBMS + an LLM can work together:

- ☐ DBMS handles storage, indexing, querying, filtering
- ☐ GPT-5.1 handles interpretation, detection, reasoning, and prediction

3. Objectives

- ☐ Design a MySQL database for storing cybersecurity logs
- ☐ Insert sample data (realistic but safe)
- ☐ Write meaningful SQL queries for analysis
- ☐ Use GPT-5.1 to detect anomalies
- ☐ Compare human-style vs AI-driven analysis
- ☐ Demonstrate how AI enhances incident detection

4. ER Diagram

sql

Copy code

Ask ChatGPT

5. Database Schema

sql

Copy code

```sql
CREATE TABLE auth_logs (
    log_id INT AUTO_INCREMENT PRIMARY KEY,
    timestamp DATETIME,
    username VARCHAR(50),
    src_ip VARCHAR(45),
    action VARCHAR(20)
);

CREATE TABLE firewall_events (
    event_id INT AUTO_INCREMENT PRIMARY KEY,
    timestamp DATETIME,
    src_ip VARCHAR(45),
    dest_ip VARCHAR(45),
    action VARCHAR(50),
    rule_triggered VARCHAR(100)
);

CREATE TABLE network_traffic (
    id INT AUTO_INCREMENT PRIMARY KEY,
    timestamp DATETIME,
    src_ip VARCHAR(45),
    dest_ip VARCHAR(45),
    bytes_sent INT,
    bytes_received INT,
    protocol VARCHAR(10)
);

CREATE TABLE threat_intel (
    threat_id INT AUTO_INCREMENT PRIMARY KEY,
    ip_address VARCHAR(45),
    threat_type VARCHAR(50),
    severity VARCHAR(20)
);
```

6. Sample Data Insertion

sql Copy code

```sql
INSERT INTO auth_logs (timestamp, username, src_ip, action) VALUES
('2025-12-01 10:22:12', 'admin', '192.168.1.50', 'FAILED'),
('2025-12-01 10:22:14', 'admin', '192.168.1.50', 'FAILED'),
('2025-12-01 10:22:16', 'admin', '192.168.1.50', 'FAILED'),
('2025-12-01 10:25:20', 'root', '203.0.113.45', 'FAILED'),
('2025-12-01 10:25:22', 'root', '203.0.113.45', 'FAILED'),
('2025-12-01 10:29:00', 'john', '10.0.0.25', 'SUCCESS');

INSERT INTO firewall_events (timestamp, src_ip, dest_ip, action, rule_triggered) VALUES
('2025-12-01 09:10:00', '203.0.113.45', '192.168.1.10', 'BLOCKED', 'Brute-force protectio
('2025-12-01 09:12:30', '45.55.10.8', '192.168.1.10', 'BLOCKED', 'Port scan detected'),
('2025-12-01 11:00:00', '10.0.0.25', '8.8.8.8', 'ALLOWED', 'DNS request');

INSERT INTO threat_intel (ip_address, threat_type, severity) VALUES
('203.0.113.45', 'Brute-force attacker', 'High'),
('45.55.10.8', 'Port scanner', 'Medium');
```

7. SQL Queries for Analysis

7.1 Detection repeated Authentication Failures

```sql
INSERT INTO auth_logs (timestamp, username, src_ip, action) VALUES
('2025-12-01 10:22:12', 'admin', '192.168.1.50', 'FAILED'),
('2025-12-01 10:22:14', 'admin', '192.168.1.50', 'FAILED'),
('2025-12-01 10:22:16', 'admin', '192.168.1.50', 'FAILED'),
('2025-12-01 10:25:20', 'root', '203.0.113.45', 'FAILED'),
('2025-12-01 10:25:22', 'root', '203.0.113.45', 'FAILED'),
('2025-12-01 10:29:00', 'john', '10.0.0.25', 'SUCCESS');

INSERT INTO firewall_events (timestamp, src_ip, dest_ip, action, rule_triggered) VALUES
('2025-12-01 09:10:00', '203.0.113.45', '192.168.1.10', 'BLOCKED', 'Brute-force protection'),
('2025-12-01 09:12:30', '45.55.10.8', '192.168.1.10', 'BLOCKED', 'Port scan detected'),
('2025-12-01 11:00:00', '10.0.0.25', '8.8.8.8', 'ALLOWED', 'DNS request');

INSERT INTO threat_intel (ip_address, threat_type, severity) VALUES
('203.0.113.45', 'Brute-force attacker', 'High'),
('45.55.10.8', 'Port scanner', 'Medium');

# 7.1 Detecting repeated authentication failures
SELECT src_ip, COUNT(*) AS attempts
FROM auth_logs
WHERE action = 'FAILED'
GROUP BY src_ip
HAVING attempts > 2;
```

100% 20:61

Result Grid Filter Rows: Search Export:

src_ip	attempts
192.168.1.50	3

7.2 Checking if failed IPs exist in threat intel

```sql
# 7.2 Checking if failed IPs exist in threat intel
SELECT a.src_ip, t.threat_type, t.severity
FROM auth_logs a
JOIN threat_intel t
ON a.src_ip = t.ip_address
WHERE a.action = 'FAILED';
```

100% 1:62

Result Grid Filter Rows: Search Export:

src_ip	threat_type	severity
203.0.113.45	Brute-force attacker	High
203.0.113.45	Brute-force attacker	High

7.3 Most suspicious IP based on cross-table activity

```sql
# 7.3 Most suspicious IP based on cross-table activity
SELECT a.src_ip, COUNT(a.log_id) AS auth_failures, COUNT(f.event_id) AS fw_events
FROM auth_logs a
LEFT JOIN firewall_events f ON a.src_ip = f.src_ip
GROUP BY a.src_ip;
```

100% 6:56

Result Grid Filter Rows: Search Export:

src_ip	auth_failur...	fw_events
192.168.1.50	3	0
203.0.113.45	2	2
10.0.0.25	1	1

7.4 High traffic anomalies

```sql
# 7.4 High traffic anomalies
SELECT src_ip, SUM(bytes_sent + bytes_received) AS total_bytes
FROM network_traffic
GROUP BY src_ip
HAVING total_bytes > 5000000;
```

100% 1:61

Result Grid Filter Rows: Search Export:

src_ip	total_bytes

8. Using GPT-5.1 for Threat Analysis

8.1 Prompt Used

“Analyze this authentication log summary and firewall correlation. Identify potential attacks, threat severity, and MITRE ATT&CK mapping. Data: [paste SQL output]”

8.2 GPT-5.1 Output (Raw)

IP 203.0.113.45 shows classic credential stuffing / brute-force behavior.
MITRE ATT&CK: T1110 – Brute Force

The same IP also hits the firewall and gets blocked → “High confidence malicious.”

IP 45.55.10.8 scanned multiple ports.
MITRE ATT&CK: T1046 – Network Service Scanning

Internal user “john” shows normal activity.

Repeated admin authentication failures from LAN IP 192.168.1.50 may suggest:
user forgetting password
malware attempting lateral movement

Recommended action:
Add IP 203.0.113.45 to blocklist permanently
Enable rate-limiting on SSH
Review endpoint at 192.168.1.50

9. AI-Driven Insights (Summary)

GPT-5.1 is surprisingly good at:

Explaining logs in plain English
Identifying attacker behavior patterns
Assigning severity
Mapping attacks to MITRE ATT&CK
Suggesting remediation steps
Spotting correlations across tables

Traditional SQL can show patterns,
but AI explains the meaning behind those patterns.

10. Comparison: SQL vs GPT-5.1

Task	SQL	GPT-5.1
Count failed logins	✓	✓
Explain *why* the pattern is dangerous	✘	✓
Detect port scan behavior	✓	✓
Classify attack category	✘	✓
Predict future risk	✘	✓
Correlate across multiple logs	✓	✓
Recommend security fixes	✘	✓

SQL gives the "what".
GPT gives the "why" and "what next".

11. Conclusion

This experiment showed that combining a DBMS with AI results in a much faster and smarter security workflow.

MySQL stores the raw truth
SQL extracts facts
GPT-5.1 converts raw logs into intelligence
MITRE mapping makes the insights enterprise-grade

Analysts get faster and better incident visibility
This exact setup could scale into a mini-SOC platform.

GPT Link:
“https://chatgpt.com/share/692e8722-6398-8011-86de-9ab689e468fc**”**

—---------- END OF DOCUMENT —----------