How I Built an AI SOC Analyst Assistant with Gemini API for Cybersecurity Triage

A generative AI agent that transforms security logs into structured insights using Gemini Pro and Retrieval-Augmented Generation (RAG)

Introduction

As part of the GenAI Intensive Course Capstone 2025Q1 hosted on Kaggle, I developed an AI-powered agent to support cybersecurity analysts in Security Operations Centers (SOCs). The agent uses Google's Gemini Pro model to convert raw security logs into structured, meaningful insights - helping analysts act faster and smarter during incidents.

This project demonstrates the application of few-shot prompting, structured output, and retrieval-augmented generation (RAG) - core GenAI capabilities that drive real-world value.

Problem Statement

Security analysts face hundreds of noisy alerts each day. Parsing logs, identifying threats, mapping them to frameworks like MITRE ATT&CK, and deciding how to respond - all under time pressure - is overwhelming.

What if we had an AI assistant that could:

- Read the logs,
- Summarize them in simple language,
- Map them to attack techniques,
- Recommend what to do next?

That's what I set out to build.

Tools & Technologies

- Google Gemini Pro API (google.generativeai)
- Kaggle Secrets for secure API key handling
- FAISS + TF-IDF for vector search (RAG)
- Pandas for log handling
- Few-shot prompting for consistency

Dataset

I used a simulated CSV dataset named cybersecurity_attacks.csv, which contains realistic alert data, including:

- source_ip, destination_ip
- user, command_line, event
- timestamp, and more

Each row is treated as a single log entry for analysis.

How the AI Agent Works

1. Gemini Summarizer (Few-shot + Structured Output)

A well-crafted prompt helps the model return:

- A human-readable summary
- Mapped MITRE ATT&CK technique

- Suggested remediation steps

2. Retrieval-Augmented Generation (RAG)

I used FAISS to create a vector store of logs, then retrieved the most similar entries to any incoming log using cosine similarity. This gives Gemini context to make better decisions.

3. Agent Reasoning

The final step combines the original log and the RAG context and prompts Gemini to:

- Reason like an analyst
- Offer a bigger-picture interpretation
- Recommend incident response actions

Example Input and Output

Raw Log:

User: jsmith

Command: powershell -enc JAB...

Event: Obfuscated PowerShell command

Gemini Response:

Summary: User 'jsmith' executed a suspicious obfuscated PowerShell command.

MITRE: T1059.001 (PowerShell)

Actions: Review execution history, isolate the system, scan for persistence.

Agent Reasoning with RAG:

This activity resembles previous lateral movement attempts. Recommend isolating the host, checking Active Directory authentication logs, and reviewing potential privilege escalation behavior.

Best Practices

- No hardcoded secrets: API key securely loaded using kaggle_secrets
- Modular codebase: Easy to read, maintain, and test
- Lightweight and stateless: Suitable for real-world deployment

Limitations

- The logs are synthetic and not real-time
- Assumes consistent format and clean input
- Currently only text-based; no multimodal input yet

What's Next?

Future improvements include:

- MITRE ATT&CK JSON integration
- Visual dashboards (Streamlit or Plotly)
- Document understanding (PDF incident reports)
- Multi-modal reasoning (screenshots, configs, alerts)

Final Thoughts

The AI SOC Analyst Assistant shows how generative AI can empower cybersecurity teams - turning noisy logs into actionable intelligence.

With models like Gemini Pro and the right prompt strategy, we're not just summarizing logs - we're

accelerating security operations.

Want to try it or collaborate on next-gen SOC automation? Let's connect.

Built by [Your Name], April 2025 - for the GenAI Intensive Course Capstone on Kaggle