■■ IR-GPT: Building an AI-Driven Incident Response Assistant

By Adiwakar | Security+ Certified | M.S. Cybersecurity Risk Management, Indiana University

1. Project Overview

IR-GPT is a locally deployed AI assistant designed to support cybersecurity analysts and GRC teams in incident response. It uses Retrieval-Augmented Generation (RAG) to provide structured recommendations aligned with NIST SP 800-61 and NIST Cybersecurity Framework (CSF). The system analyzes alerts or incident descriptions, retrieves relevant playbook data, and produces clear, auditable outputs with JSON summaries and NIST mapping.

2. Architecture & Workflow

IR-GPT is built using Python, Streamlit, and ChromaDB. It operates locally with Ollama, allowing models such as Phi-3 Mini or Mistral to run offline. Documents and playbooks are vectorized using Sentence Transformers and stored in ChromaDB for retrieval. The RAG pipeline ensures that model responses are grounded in verified playbooks, minimizing hallucinations and maintaining accuracy. The workflow is straightforward: the user describes an incident, IR-GPT retrieves relevant sections from NIST-based playbooks, and the model synthesizes actionable steps.

3. Development Process

- Installed Ollama and Python (3.10+) on Ubuntu for local inference. - Created virtual environment and installed dependencies from `requirements.txt`. - Integrated Streamlit for an interactive UI where analysts can input incidents or logs. - Added multiple playbooks (MFA fatigue, insider threats, privilege escalation) to train retrieval context. - Configured the app to output narrative guidance and structured JSON summaries mapped to frameworks.

4. Data & Privacy Considerations

All inference occurs locally on-device, ensuring no data leaves the environment. No cloud APIs or telemetry are used. This makes IR-GPT compliant with on-premise and regulated environments where sensitive data must not be transmitted externally. Logs or user inputs are not stored permanently unless explicitly exported as summaries.

5. Governance & Framework Mapping

IR-GPT aligns with the NIST Cybersecurity Framework and NIST SP 800-61 Incident Response phases. - **Identify:** Classifies incident type from user input. - **Detect:** Analyzes event logs, correlates anomalies. - **Respond:** Suggests containment and mitigation actions. - **Recover:** Recommends post-incident validation. It also loosely aligns with ISO 27035 and SANS six-step IR model, providing flexible framework coverage.

6. Limitations & Human Oversight

IR-GPT is not a replacement for analysts. It assists in triage, documentation, and framework alignment. Human oversight is critical for validating containment actions and verifying remediation accuracy. It performs best on structured or described incidents but is less effective on unstructured or zero-day events.

7. Sample Usage

Example prompt: “Users report multiple failed logins followed by one successful login. Multiple permissions have been changed on host files.” IR-GPT retrieves relevant MFA fatigue and account compromise playbooks, analyzes probable causes, and outputs NIST-mapped response guidance. The structured JSON output includes incident phase, severity, containment steps, and notes for ticketing.

8. Organizational Integration

In a SOC or GRC environment, IR-GPT can serve as an internal triage or documentation assistant. It helps analysts generate standardized, framework-aligned responses faster while ensuring audit readiness. Training requirements are minimal; most users interact through the Streamlit UI. Generated JSON summaries can be integrated into ticketing systems or SIEM workflows.

9. Recruiter Q&A; Summary

How does it work? It uses a RAG architecture combining local embeddings and playbooks stored in ChromaDB, with LLM inference from Ollama models. **Fine-tuning or prompting?** Pure prompt engineering and retrieval, no fine-tuning. **Accuracy control?** Retrieval grounding, low temperature, and verified playbooks minimize hallucinations. **Data use?** Fully local, ensuring confidentiality. **Why NIST frameworks?** They’re globally accepted for IR and GRC, giving structure and credibility to outputs. **Limitations?** Lacks zero-day detection and requires human review for containment steps. **How could it be deployed?** As an internal SOC assistant, audit prep tool, or training simulator.