Redacted AI Security Audit & Compliance Assessment (Sample)

Contents

Attestation: This document is a sanitized, representative sample of an AI security audit performed on a live enterprise customer. All specific architecture endpoints, proprietary data boundaries, and API keys have been removed or replaced.

1 Executive Summary

Yugen Risk Advisors conducted a targeted **AI Security and Compliance Audit** against the **BetaCorp Generative AI Gateway and RAG Platform** (placeholder entity). This audit evaluated the security posture of BetaCorp's custom Large Language Model (LLM) orchestration layer, retrieval-augmented generation (RAG) data flows, and automated support agents.

Overall risk is rated as **Medium-High** due to a lack of isolated system prompt execution sandboxes, data leakage vectors in vector database queries, and vulnerable downstream tool integration paths.

1.1 Key Findings

- **Indirect Prompt Injection:** Downstream support agents can be coerced into execution flow hijacking when processing untrusted incoming support email transcripts containing adversarial prompts.
- **Vector Database Boundary Leaks:** Vector DB queries lack strict user-attribute scoping, allowing low-privileged authenticated users to retrieve snippets of confidential administrative documents via RAG retrieval.
- **Missing State Sandboxing:** Multi-agent DevOps swarms retain execution memory across user sessions, creating a cross-session session-state poisoning risk.

1.2 Business Impact & Recommendations

Compromise of internal orchestration workflows, unauthorized access to sensitive company files, and potential data exfiltration through agentic API execution. Remediation should focus on **session-isolated RAG scope enforcement**, **prompt boundary isolation via system-only message structures**, and **deterministic tool-parameter schema validation**.

2 Technical Summary

2.1 Scope

- Retrieval-Augmented Generation (RAG) retrieval pipeline & Vector DB boundaries (Pinecone).
- Agent orchestrator tooling & downstream API integrations (LangChain / custom Python runtimes).
- Prompt-injection threat modeling & adversarial boundary validation.
- Audit trailing and state-leakage analysis.

3 Table of Findings

ID	Title	Severity	Likelihood	Surface	Status
F-001	Indirect Prompt Injection leading to Tool Execution Hijacking	**High**	High	Support Agent Mail-Ingest	Unfixed
F-002	RAG Vector DB Context Leakage via Unscoped Tenant Queries	**High**	Medium	Pinecone Vector Database	Unfixed
F-003	Cross-Session State Poisoning in Agentic Memory	**Medium**	Low	Agent Orchestration Layer	Unfixed
F-004	Insecure Downstream Shell Tool Argument Sanitization	**Medium**	Medium	DevOps Deployment Swarm	Unfixed

4 Risk Ratings

Final categorical rating (**Critical / High / Medium / Low / Informational**) is derived from a composite score based on impact, likelihood, and surface exposure.

4.1 Dimension Criteria

- **Severity:** The direct business, compliance, or financial blast radius of successful exploit.
- **Exploitability:** The technical prerequisites and reliable tooling required to achieve compromise.
- **Exposure:** The access boundaries protecting the target surface.