Suhas Bhairav

# AI Application Threat Modeling Security Checklist

## A practical workflow to audit architecture with LLM security checks

Deliver a repeatable, low-friction LLM-driven security review of architectures.

Suhas Bhairav

suhasbhairav.com

CONTEXT

Engineering leaders and teams

# LLM Threat Modeling Workflow

This page defines the end-to-end flow for feeding software architecture designs into an LLM to surface security issues. It establishes data, tools, roles, and governance required to act on findings.

**Key move** **Start with a scoped pilot to learn before scaling.**

## PRACTICAL CHECKLIST

| | |
|---|---|
| 01 | Define architecture artifacts to analyze: diagrams, design docs, API specs. |
| 02 | Choose an LLM with strong security evaluation capabilities and data handling. |
| 03 | Specify data boundaries and access controls for inputs/outputs. |
| 04 | Map findings to OWASP categories and business risk. |
| 05 | Integrate with your CI/CD to run checks on PRs. |
| 06 | Define human review steps for ambiguous results. |
| 07 | Assign ownership and a remediation SLA per finding. |

**Suhas Bhairav AI Resource**

Deliver a repeatable, low-friction LLM-driven security review of architectures.

# Inputs, Tools, Guardrails

This page details what to feed the LLM, which tools to use, and how to structure guardrails. It aligns architecture outputs with security expectations.

**Key move** **Limit LLM hallucinations via verification checkpoints.**

## PRACTICAL CHECKLIST

| | |
|---|---|
| **01** | Architecture artifacts: diagrams, API specs, data models, threat models. |
| **02** | Data handling: remove PII, anonymize data, and limit sensitive inputs. |
| **03** | LLM prompts: define roles, goals, and expected outputs per artifact. |
| **04** | Tools: choose a model hosting option (on-prem, cloud) with audit logs. |
| **05** | Security guardrails: require human-in-the-loop for high-severity findings. |
| **06** | Data flow: map artifacts to inputs and LLM outputs to actionable findings. |
| **07** | Compliance: ensure logs, retention, and access controls meet policy. |

**Suhas Bhairav AI Resource**

Deliver a repeatable, low-friction LLM-driven security review of architectures.

# Adoption, Metrics, Ownership

Outline a weekly, low-friction adoption path for engineering teams. Define metrics to gauge risk reduction and process maturity.

**Key move** **Maintain explicit approvals for high-severity findings.**

## PRACTICAL CHECKLIST

| | |
|---|---|
| 01 | Week 1: establish a pilot scope with one service and one data surface. |
| 02 | Week 2: capture baseline security findings from existing designs. |
| 03 | Week 3: integrate LLM checks into PR reviews and pipelines. |
| 04 | Week 4: assign owners, remediation SLAs, and escalation paths. |
| 05 | Risks & guardrails: identify core risks and define mitigations. |
| 06 | Measurement: track findings closed rate, time-to-remediation, and reoccurrence. |
| 07 | Ownership: appoint a threat modeling steward and cross-functional review board. |

**Suhas Bhairav AI Resource**

Deliver a repeatable, low-friction LLM-driven security review of architectures.

ABOUT THE CREATOR

# Suhas Bhairav

I build AI systems that combine distributed architecture, retrieval, knowledge graphs, security, and practical workflow design. My focus is AI that can be used repeatedly by real teams: understandable, reliable, and grounded in the work people already do.

## EXPERTISE

- Principal systems architecture and applied AI engineering
- Production-grade AI systems, agentic workflows, RAG, and knowledge graphs
- Full-stack workflow applications across AWS, Google Cloud, Azure, on-premise, and hybrid environments
- Practical AI implementation patterns for automotive, manufacturing, finance, logistics, cybersecurity, sales, and customer operations
- Security, governance, observability, approvals, and failure-mode thinking for operational AI systems


WEBSITE

**suhasbhairav.com**

EMAIL

**suhasbhairav@gmail.com**