AI SECURITY ENGINEERING FIELD GUIDE · 2026

Chapter 02 · LLM Application Security

Standalone learning module for LMS delivery and required reading.

FORMAT	USE	SCOPE	AUDIENCE
Standalone PDF	LMS reading module	Single chapter	Learners

CHAPTER 02

LLM Application Security

> An LLM app is still an application. Review the glue code, not only the model.

> An LLM app is still an application. Review the glue code, not only the model.
>
> FIELD GUIDE

> **FIELD GUIDE: APPLICATION REVIEW**
>
> Map the request path, prompt assembly, provider boundary, output handling, authorization decisions, and logs before testing model behavior.

This domain covers LLM-powered application surfaces: prompts, orchestration code, API calls, authentication, authorization, session handling, secrets, output rendering, plugins, extensions, and provider boundaries. It matters when product teams ship copilots, chat interfaces, summarizers, classifiers, assistants, or AI features inside existing applications.

Treat the review like an AppSec review with a new execution layer. The model call is only one step in the request path. The surrounding application decides what data enters the prompt, what authority is available, what output can trigger, and what evidence survives.

> **FIELD USE**
>
> Start at the user action, then follow the request through prompt assembly, model call, output handling, and any downstream effect. Mark a release blocker when authorization, policy enforcement, or state change depends on model judgment alone.

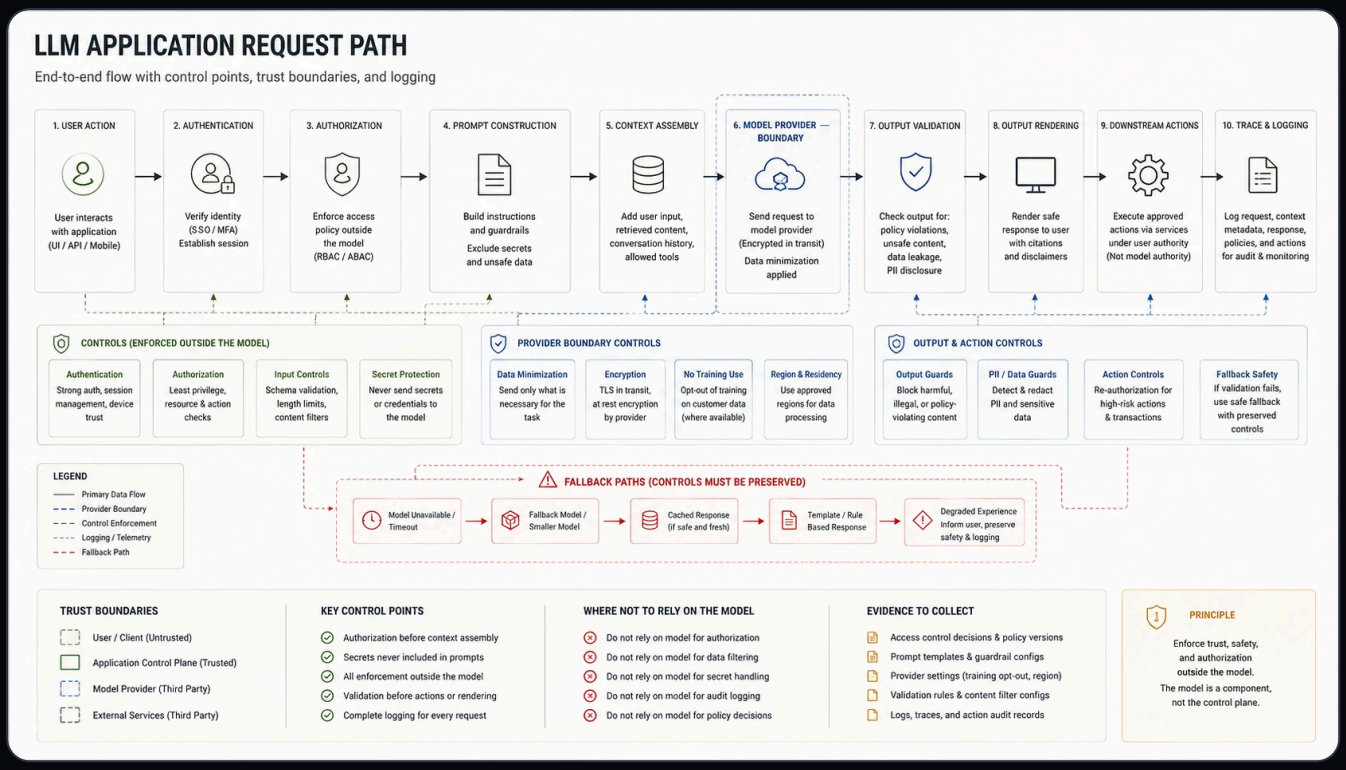

FIGURE 1: FIGURE 3: LLM APPLICATION REQUEST PATH FROM USER ACTION THROUGH PROMPT ASSEMBLY, MODEL CALL, OUTPUT HANDLING, AND LOGGING

WHAT BREAKS HERE?

- Secrets, API keys, hidden instructions, or debug state enter prompts.
- The model is asked to enforce authorization, entitlement, policy, or tenancy.
- Output is rendered into unsafe channels without encoding, schema checks, or review.
- Prompt and output logs capture sensitive data without access rules or retention limits.
- Provider calls create data exposure, credential, continuity, or contractual gaps.
- Fallback paths skip the controls that the primary AI path depends on.

WHAT TO INSPECT

- Prompt templates, system messages, prompt construction code, and context assembly.
- Authentication and authorization checks before model, retrieval, provider, and tool calls.
- Provider SDK usage, API keys, retry paths, rate limits, and error handling.
- Output rendering paths into HTML, markdown, tickets, emails, code, or workflow systems.
- Logs, traces, analytics events, and support tooling that store prompts or outputs.
- Feature flags, fallback routes, cached responses, and manual override paths.

WHAT TO ASK

- What authority does the application have before the model is called?
- Which inputs are untrusted, user-controlled, retrieved, or tool-generated?
- Can output trigger action, create records, send messages, or influence decisions?
- Which checks happen before the model sees data, and which checks happen after output?
- Where are provider terms, data retention settings, and training-use settings recorded?
- Which failures block release, and which create backlog items?

WHAT TO TEST

- Submit malicious markdown, links, HTML, JSON, code blocks, and instruction conflicts.
- Try low-privilege requests for high-privilege summaries, records, and workflow actions.
- Check whether secrets or hidden instructions are exposed through prompt echo or debug output.
- Force provider errors, timeout paths, fallback paths, and malformed outputs.
- Replay the same request across roles, tenants, feature flags, and fallback modes.
- Verify logs contain enough evidence without storing unnecessary sensitive data.

CONTROLS AND GUARDRAILS

- Authorization before model calls, retrieval calls, provider calls, and downstream actions.
- Prompt construction standards with secret exclusion and context labeling.
- Output encoding, schema validation, content policy checks, and action separation.
- Provider boundary controls for keys, retention, training use, and failure behavior.
- AI trace logging that captures request IDs, model, prompt version, policy decision, and output handling.
- Release gates for prompt changes, provider changes, new output sinks, and new actions.

EVIDENCE TO COLLECT

- Prompt template inventory.
- Application data-flow map.
- Authorization test results.
- Provider configuration notes.
- Output validation test evidence.
- Logging and retention policy.
- Fallback control notes.
- Remediation backlog.

OUTPUT ARTIFACTS

- LLM application security review.
- Prompt and context assembly map.
- Provider boundary statement.
- Secure output handling checklist.
- Release gate findings.

RELATED SERVICES AND WORKBENCH TOOLS

TYPE	RELATED PATHS
Workbench	Surface Scanner, Runtime Proxy, Threat Canvas
Services	AI Product Security Assessment, Product Security Baseline, Enterprise AI Security Readiness
Handbook	Architecture and Trust Boundaries, Prompt Injection, Model and Provider Risk

ARTIFACT: LLM APP REVIEW PACK

Produce a prompt map, provider boundary statement, authorization notes, output tests, and logging evidence.