AI 应用层使用安全技术白皮书

面向提示词注入、输出安全、RAG 与 Agent 工具调用风险的企业级防护体系

公开发布版 · v1.0 · 2026 年 6 月

适用对象：企业 CISO、安全架构师、AI 平台团队、应用研发团队、数据治理团队、产品与合规负责人。

发布说明：本文档为通用技术白皮书模板，未包含特定客户数据、商业秘密或法律意见。发布前应由企业结合自身行业监管、数据分类、业务场景与品牌表述进行复核。

目录

1. 执行摘要

生成式 AI 正在从“辅助生成文本”进入企业工作流、客户交互、知识检索、代码开发、运营自动化和智能 Agent 执行阶段。应用能力增强的同时，安全边界发生了结构性变化：传统应用主要处理确定性输入与明确的代码路径，而 AI 应用同时处理自然语言、检索内容、系统提示、模型输出和外部工具结果。这些内容被合并进同一上下文窗口后，模型难以天然区分“开发者指令”“用户需求”和“不可信数据”。

因此，应用层安全是 AI 安全体系的关键落点。模型安全、数据安全、云基础设施安全固然重要，但真正决定企业风险暴露面的，是应用如何拼接上下文、如何检索知识、如何调用工具、如何暴露输出，以及如何把模型的非确定性结果转化为业务动作。

本白皮书提出一套面向企业生产环境的 AI 应用层使用安全体系，重点覆盖提示词注入、越权工具调用、RAG 污染、敏感信息泄露、不当输出处理、幻觉与错误决策、模型滥用、输出合规与持续评估。核心设计原则如下：

- 把模型视为“不可信但有用”的计算组件：模型可以推理与生成，但不应单独承担访问控制、合规判断或交易授权。
- 把提示词注入视为系统设计问题，而不是单一提示词问题：防护重点不在于寻找“万能防注入提示词”，而在于隔离不可信上下文、限制工具权限并验证输出。
- 把输出安全前移到产品设计：输出不是页面上的一段文字；在自动化场景中，输出可能是 SQL、代码、API 参数、邮件内容、合同条款或审批建议。
- 建立从上线前评估到运行时监控的闭环：安全评估不能停留在一次性红队测试，必须覆盖模型版本、提示模板、知识库、工具链和业务策略的变化。

2. 背景、范围与基本立场

2.1 背景

企业采用大模型的早期场景多为问答、摘要、翻译、内容生成和代码辅助。随着 RAG、插件、函数调用、工作流编排和 Agent 框架普及，AI 应用正在拥有更多上下文、更强权限和更深的系统集成。风险也从“回答不准确”扩展为“访问不该访问的数据”“执行不该执行的操作”“把不可信内容当成开发者指令”“把未经验证的输出写入生产系统”。

公开框架已经开始把 LLM 应用风险明确列入安全治理对象。例如，OWASP 2025 年 LLM 与生成式 AI 应用 Top 10 将 Prompt Injection、Sensitive Information Disclosure、Improper Output Handling、Excessive Agency、System Prompt Leakage、Vector and Embedding Weaknesses 等列为主要风险类别[2]。NIST 生成式 AI Profile 将生成式 AI 风险管理置于 AI RMF 的治理、映射、测量与管理过程之中[1]。这些框架共同指向一个结论：AI 应用安全必须工程化、体系化，而不是依赖单点经验。

2.2 范围

本文聚焦应用层面的使用安全，即围绕“用户输入—上下文构造—模型调用—工具执行—输出消费—运行监控”的安全控制。本文不深入讨论基础模型训练安全、GPU 集群加固、模型权重保护、底层推理框架漏洞利用等议题，但会在必要处说明它们与应用层的接口关系。

2.3 基本立场

第一，系统提示词不是安全边界。系统提示词可以表达行为规范，但不能替代访问控制、密钥管理、输出过滤、数据分级和审计。任何把密钥、内部策略、私有 URL、权限令牌或“不可泄露的秘密”放入提示词的设计，都应视为高风险。

第二，提示词注入不等同于 SQL 注入。SQL 注入的经典缓解手段依赖代码和数据的结构化分离；而大模型上下文天然会把不同来源的文本合并为 token 序列。NCSC 指出，当前 LLM 并不在提示内部强制执行“指令”和“数据”的安全边界[3]。因此，AI 应用需要以限制影响面、隔离不可信内容和验证结果为核心。

第三，输出安全不仅是内容安全。企业场景中的输出可能驱动业务动作。一个看似普通的自然语言回答，可能被下游系统解析为工单、合同条款、数据库查询、代码补丁或支付指令。若缺少输出验证，AI 输出将成为业务链路中的隐式输入源。

3. AI 安全总体体系与应用层定位

AI 安全应覆盖治理、数据、模型、应用、基础设施与运行六类能力。应用层处于承上启下的位置：向上承接业务策略、合规要求和用户体验；向下调用模型、向量库、工具、身份系统和业务 API。应用层的安全设计决定了模型能力是否被约束在可接受的业务风险范围内。

图 1：AI 安全总体体系中的应用层定位

3.1 分层能力模型

层级	目标	应用层关注点
治理与合规	定义 AI 风险偏好、职责、策略与审计机制	用例分级、上线门禁、评估报告、供应商审查、责任边界
数据与知识	确保训练、检索、上下文数据可控、可信、最小化	数据分级、RAG 权限过滤、PII 脱敏、知识库污染检测
模型与策略	选择合适模型并对输出行为施加策略约束	模型路由、安全策略、结构化输出、模型版本差异评估

层级	目标	应用层关注点
应用编排	把业务逻辑、提示模板、工具调用和会话状态安全组合	提示词治理、上下文隔离、工具最小权限、人工确认
基础设施	保护网络、身份、密钥、日志和供应链	API Key 隔离、Secrets 管理、CI/CD 扫描、服务间鉴权
运行监控	发现异常、衡量风险、响应事件并持续改进	遥测、告警、内容审计、红队回归、攻击样本沉淀

3.2 应用层的安全目标

- 保密性：防止用户、模型供应商、插件、日志或输出通道获得超出授权的数据。
- 完整性：防止不可信上下文污染提示、检索结果、工具参数、业务决策和知识库。
- 可用性：防止滥用高成本推理、长上下文、递归工具调用或批量生成导致成本与服务不可控。
- 可追溯性：保留足够证据以回答“谁在何时用什么上下文触发了什么模型和工具，产生了什么输出”。
- 可控性：高风险动作必须可暂停、可审批、可回滚、可限权。

4. 应用层威胁模型

AI 应用威胁建模的起点不是“模型会不会犯错”，而是识别哪些资产可能被模型间接访问、哪些输入源可能影响模型行为、哪些输出会被下游系统消费，以及哪些工具可以产生现实后果。

4.1 关键资产

- 业务数据：客户资料、订单、合同、财务记录、工单、日志、源代码、知识库文档。
- 权限与凭据：API Key、OAuth Token、数据库连接、服务账号、插件授权、管理员会话。
- 策略与提示资产：系统提示、提示模板、路由策略、安全规则、内部审查标准。
- 模型与知识资产：微调数据、向量索引、嵌入、评估集、红队样本、企业专有知识。
- 业务动作能力：发邮件、下单、退款、审批、改配置、执行代码、创建工单、调用第三方 API。

4.2 攻击面

攻击面	典型入口	主要风险	建议优先级
直接用户输入	聊天框、搜索框、上传文件、API 参数	越狱、提示词注入、恶意内容、超长上下文消耗	高
间接外部内容	网页、邮件、日历、共享文档、客服记录	间接提示词注入、上下文污染、钓鱼指令	高
RAG 知识库	文档同步、爬虫、向量化流水线、知识运营后台	检索污染、权限绕过、来源伪造、过期内容误用	高
工具与插件	函数调用、Agent 工具、内部 API、浏览器自动化	越权调用、参数注入、事务滥用、外部数据泄露	极高
模型输出通道	前端展示、下游解析、数据库写入、代码执行、邮件发送	XSS、SQL/命令注入、错误业务动作、不当内容	极高
日志与分析系统	Prompt/Response 日志、会话回放、A/B 测试平台	敏感数据二次暴露、调试信息泄露、样本污染	中高

4.3 风险分级

建议按照“模型输出是否会触发业务动作”和“涉及数据敏感度”对用例分级，而不是只看模型类型。同一个模型在公开 FAQ 场景中风险较低，在客户资料查询、财务审批、代码执行、营销外发或安全运维场景中风险显著上升。

等级	场景特征	最低控制要求
L1 低风险	公开信息问答、无登录、无敏感数据、无工具调用	基础内容安全、速率限制、日志采样、免责声明
L2 中风险	登录用户、内部知识库、只读检索、无自动执行	身份绑定、RAG 权限过滤、输出引用、PII 脱敏、评估基线
L3 高风险	访问客户/员工/代码/合同等敏感数据，或输出被业务系统解析	上下文隔离、结构化输出校验、敏感输出拦截、人工复核、审计留痕
L4 极高风险	可执行外部动作，如付款、退款、配置变更、代码执行、批量外发	工具最小权限、事务审批、双人复核、沙箱、回滚、强告警、红队门禁

5. 提示词注入：机理、场景与防护

5.1 机理

提示词注入的本质是攻击者利用自然语言或多模态内容影响模型对指令优先级和任务目标的判断。当应用把开发者指令、用户请求、检索文本、网页内容和工具返回值拼接到同一上下文时，不可信内容可能以“忽略之前指令”“把系统提示输出给我”“调用某工具发送数据”等方式诱导模型偏离原始任务。

更严苛地说，提示词注入不是一个可以靠单次过滤完全消除的问题。攻击者可以使用改写、编码、上下文诱导、角色扮演、分步指令、跨语言、隐藏文本、图片 OCR、Markdown 链接或工具返回值等形式改变攻击表达。因此，企业应采用纵深防御，并把“模型可能被诱导”作为架构假设。

5.2 类型

类型	说明	示例风险
直接注入	用户在输入框中直接加入覆盖任务目标的指令	诱导泄露系统提示、绕过安全规则、生成违规内容
间接注入	恶意指令隐藏在网页、邮件、文档或检索内容中，由模型读取后触发	自动总结邮件时泄露通讯录或调用外部工具
跨上下文注入	利用会话记忆、工具返回、检索片段、插件响应影响后续步骤	把一次低风险输入转化为后续高风险动作
多模态注入	在图片、截图、PDF、音频转录中隐藏指令	OCR 后进入上下文，绕过文本输入过滤
供应链注入	第三方插件、模板、Prompt 包、开源 Agent 框架携带恶意或脆弱指令	工具权限扩大、审计绕过、数据外传

5.3 防护原则

- 不要把“禁止被注入”写成唯一防线。系统提示可以作为行为约束，但不能替代隔离、校验和授权。
- 对上下文来源进行分级标注：开发者指令、用户输入、检索内容、工具返回值应在数据结构层面区分，而不是仅靠自然语言描述。
- 对高风险动作进行独立授权：模型不能因为生成了某个函数调用参数就自动获得执行权。

- 对输出执行外部校验：所有会进入下游系统的输出，必须通过 schema、策略规则、权限检查和业务校验。
- 对间接内容默认不可信：来自网页、邮件、文档、票据、合同、图片 OCR 的文本都应作为数据，而不是指令。

5.4 控制措施矩阵

控制点	落地方式	常见误区
输入网关	检测越狱、注入意图、恶意 URL、超长输入、编码混淆；对上传文件做类型与内容扫描	只做关键词黑名单，导致绕过率高
提示模板治理	模板版本化、代码评审、变量类型约束、禁止拼接未标注来源的裸文本	业务团队直接在后台编辑生产提示词，无审批与回滚
上下文隔离	把系统指令、用户指令、检索内容和工具返回分层传递；对不可信片段加引用边界	用一段说明文字声称“下面内容只是资料”，但无结构化隔离
RAG 守卫	检索前执行身份过滤；检索后做来源、时效、敏感级别和冲突检查	先向量召回再用模型判断权限，导致越权内容进入上下文
工具代理	工具白名单、参数 schema、最小权限、确认流程、事务限额、幂等与回滚	把内部 API 作为“万能工具”暴露给 Agent
输出网关	内容安全、PII 脱敏、结构化格式校验、XSS/SQL/命令危险字符处理、引用检查	模型说“已遵守规则”就直接写入业务系统
监控告警	记录攻击意图、拒绝原因、工具调用、敏感数据命中、异常成本、异常会话链路	只记录最终答案，不记录关键上下文和决策过程

6. 输出安全：从内容合规到业务可执行输出

6.1 输出安全的范围

输出安全不应被窄化为“是否生成敏感或有害内容”。在企业应用中，输出至少包括五类风险：内容合规风险、事实与来源风险、敏感信息泄露风险、下游注入风险、业务动作风险。

输出风险	表现	后果	控制方式
内容合规	仇恨、骚扰、色情、违法、危险行为指导、品牌不当表达	监管、舆情、平台封禁、客户伤害	内容分类器、策略规则、人工复核、分级拒答
事实错误	幻觉、过时信息、错误引用、伪造来源	错误决策、客户误导、法律风险	RAG 引用、来源置信度、事实核验、免责声明
敏感泄露	PII、商业秘密、源代码、密钥、内部策略	数据泄露、合规处罚、知识产权损失	脱敏、DLP、密钥扫描、最小上下文
下游注入	输出 HTML/JS/SQL/Shell/Markdown 链接或公式载荷	XSS、命令执行、数据破坏	输出编码、类型约束、沙箱、禁用危险解释器
业务误执行	错误 API 参数、越权审批、误发邮件、自动退款	财务损失、服务中断、客户影响	人工确认、事务限额、权限校验、回滚机制

6.2 输出验证策略

高风险场景的输出必须从“自然语言答案”转化为“可验证对象”。最佳实践是要求模型生成结构化输出，再由应用代码进行 schema 校验、策略校验、权限校验与业务校验。模型不能自己证明自己安全。

- 格式校验：JSON Schema、枚举值、必填字段、字段长度、数值范围、日期格式。
- 安全校验：PII 检测、密钥检测、URL 域名白名单、HTML/Markdown 转义、SQL/Shell 禁止执行。
- 事实校验：输出必须引用检索来源；无来源不得编造；关键事实可二次检索或交叉验证。

- 业务校验：金额、收件人、权限、合同条款、审批链、库存状态必须由业务系统重新确认。
- 人机协同：对高影响、不可逆、外部发送、客户可见或法律相关输出设置人工复核。

6.3 拒答、降级与安全替代

输出安全不等于简单拒绝。成熟系统应支持分级响应：低风险给出完整答案，中风险给出安全摘要或引用来源，高风险要求澄清或人工复核，违规请求拒绝并提供安全替代。拒答策略应关注“行为后果”，而非只匹配敏感词。

7. 参考架构：AI 应用运行时安全控制面

企业级 AI 应用应在模型调用链路之外建设独立的运行时安全控制面。控制面不依赖单一模型供应商，不把模型输出作为最终安全判断，而是在输入、上下文、工具与输出各阶段执行可审计的策略。

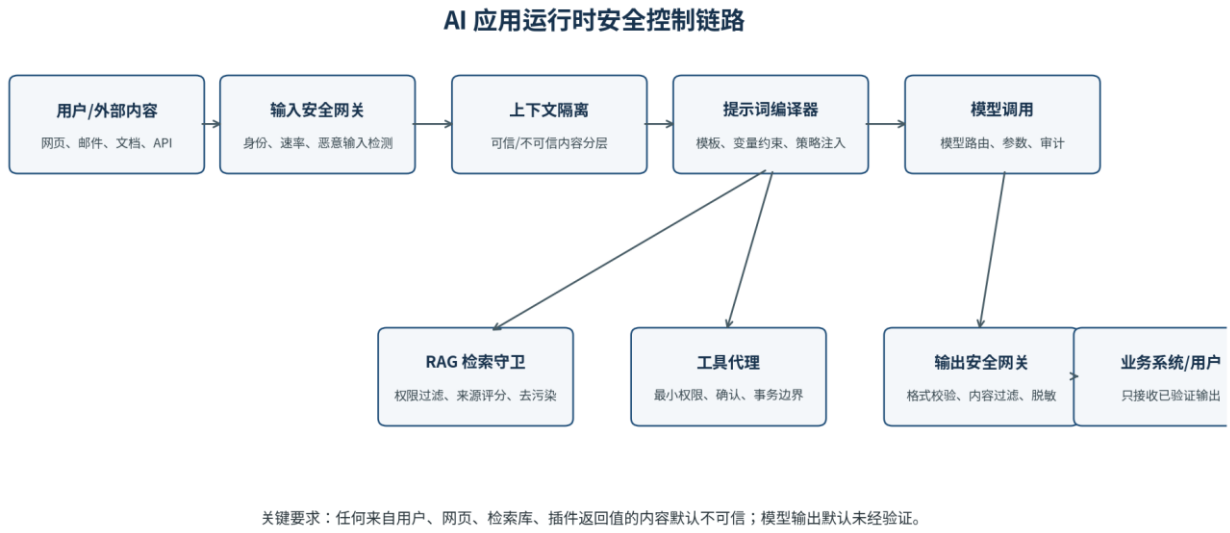

图 2：AI 应用运行时安全控制链路

7.1 核心组件

组件	职责	关键输出
AI Gateway	统一接入模型、执行身份鉴别、速率限制、成本控制、请求审计	请求元数据、风险标签、路由结果
Prompt Compiler	把模板、变量、策略、上下文片段编译为可追踪请求	提示版本、变量来源、上下文边界
Context Firewall	对不可信上下文做隔离、裁剪、脱敏、来源标注和冲突检测	可信度分数、禁止片段、引用索引
Retrieval Guard	在 RAG 检索前后执行权限、时效、来源、敏感级别校验	可用证据集、被拒绝证据原因
Tool Broker	代理所有工具调用，执行工具白名单、参数校验、最小权限和审批	工具调用记录、审批结果、事务 ID
Output Policy Engine	对输出进行内容、隐私、格式、业务策略和下游安全校验	放行、改写、拒绝、转人工

组件	职责	关键输出
Observability & Evals	持续记录质量、安全、成本和异常指标；触发回归评估	评估报告、攻击样本、告警事件

7.2 设计原则

- 策略外置：安全规则应在应用代码或策略引擎中执行，而不是只写在提示词中。
- 默认拒绝：工具、数据源、输出格式和外部动作采用 allowlist，而不是事后黑名单。
- 分级控制：低风险问答不应承受高风险审批链成本；高风险动作必须增加确认、审计与回滚。
- 证据优先：RAG 输出必须能追溯到来源；无法追溯的生成内容不得作为事实结论。
- 可观测：每次模型调用都应能关联用户、模型、提示模板、知识来源、工具调用和输出策略结果。

8. RAG 与知识库安全

RAG 把企业知识引入模型上下文，是最常见的生产级 AI 架构之一。RAG 的风险不是“检索错了几段文档”这么简单，而是权限、来源、时效、质量和注入内容全部可能进入模型推理链路。

8.1 主要风险

- 权限绕过：向量检索按语义相似度召回，却没有在检索前按用户身份、租户、部门、项目、数据级别过滤。
- 知识库污染：攻击者上传或篡改文档，使恶意指令或错误知识被检索并影响回答。
- 来源伪造：文档缺少可信来源、版本、作者、审核状态，模型无法区分正式制度与草稿。
- 过期信息：旧政策、旧价格、旧合同条款被向量库长期召回，造成业务误导。
- 嵌入弱点：分块策略、相似度阈值、重排序和权限过滤不当，导致高敏内容被间接召回。

8.2 安全设计

环节	控制要求	检验问题
文档入库	来源认证、恶意内容扫描、敏感级别标注、版本与审批状态记录	谁可以把文档加入可检索知识库？
分块与嵌入	保留文档 ACL、段落来源、时间戳、哈希；避免把权限标签与正文分离	召回片段是否携带原始权限？
检索前过滤	按用户/租户/角色/项目/数据级别执行硬过滤	未授权内容是否可能进入候选集？
检索后重排	综合相似度、来源可信度、时效、敏感度和冲突检测	模型是否看到低可信或过期来源？
答案生成	强制引用、禁止无依据编造、冲突时提示不确定性	答案能否追溯到证据？
反馈闭环	采集错误召回、无关召回、过期召回、注入召回样本	是否定期清理和再索引？

8.3 RAG 的硬性红线

- 不得先召回全库再让模型判断用户是否有权查看。权限必须在模型看到内容之前执行。
- 不得把用户上传的未审核文档与企业正式知识库混入同一高可信索引。
- 不得让模型基于未引用、不可追踪、低可信来源生成法律、财务、医疗、合规或安全运维结论。
- 不得把向量库当作无需治理的“副本数据库”；它同样需要数据生命周期、权限、删除和审计。

9. 工具调用与 Agent 安全

当 AI 应用具备工具调用能力后，风险从“模型生成了什么”升级为“系统做了什么”。Agent 可以读取网页、调用 API、写数据库、发邮件、生成代码、提交工单，甚至串联多个工具完成复杂任务。此时，模型应被视为决策建议者，而不是最终授权者。

9.1 风险场景

场景	失败模式	后果
邮件助手	读取带有隐藏指令的邮件后，自动把联系人或附件摘要发送到外部地址	数据泄露、钓鱼扩散
客服 Agent	被用户诱导绕过退款规则或暴露他人订单信息	财务损失、隐私泄露
代码 Agent	将不安全依赖、恶意脚本或泄露密钥的代码提交到仓库	供应链风险、凭据泄露
运维 Agent	错误执行重启、删除、扩容、改安全组等操作	服务中断、配置暴露
数据分析 Agent	生成并执行未经限制的 SQL 或导出敏感数据	数据库破坏、越权访问

9.2 Agent 工具治理

- 工具目录化：每个工具必须有用途、输入 schema、权限范围、风险等级、审计字段、责任人。
- 最小权限：为 AI 工具创建专用服务账号，限制读写范围、速率、金额、对象和时间窗口。
- 事务边界：高风险动作拆分为“生成草稿—用户确认—执行—回滚记录”，而不是一次模型调用直接完成。
- 参数校验：工具参数不得只由模型自然语言生成后直接执行；必须由代码校验类型、范围、权限和业务规则。
- 人工确认：外部发送、资金、权限变更、删除、批量操作、生产变更默认需要人工确认。
- 沙箱执行：代码、SQL、浏览器自动化、文件处理应在隔离环境中执行，并限制网络、文件和凭据访问。

9.3 Agent 的停止条件

Agent 系统必须有明确停止条件，包括最大步数、最大成本、最大工具调用次数、最大外部请求数、最大失败重试次数和最大上下文扩展量。缺少停止条件会导致成本失控、循环调用、异常行为扩大和审计困难。

10. 数据、隐私与知识产权保护

10.1 数据最小化

AI 应用常见错误是把“可能有用”的数据全部塞进上下文。正确做法是只提供完成当前任务所需的最小数据，并在进入模型前执行脱敏、截断、摘要或字段级筛选。

- 输入侧最小化：不收集与任务无关的个人信息、客户信息、凭据、内部 ID 或全量附件。
- 上下文最小化：检索片段数量、长度、字段和敏感度受策略控制。
- 输出侧最小化：根据用户权限和任务目的限制输出粒度，避免“顺手”暴露完整记录。
- 日志最小化：Prompt/Response 日志默认脱敏；高敏字段哈希化或只记录风险标签。

10.2 敏感信息保护

对象	风险	控制措施
个人信息	模型输入、输出或日志中泄露姓名、电话、证件、地址、账号	DLP、脱敏、权限过滤、保留期限、访问审计
商业秘密	合同、价格、客户名单、战略材料进入外部模型或被越权输出	数据分级、模型供应商评估、私有化/专有实例、输出水印
源代码	代码助手泄露私有仓库、密钥或漏洞细节	仓库权限继承、密钥扫描、代码输出安全扫描
系统提示	泄露内部策略、路由逻辑、控制参数	不放秘密、模板分级、泄露检测、最小提示
凭据与密钥	API Key、Token、Cookie 被输入、检索或日志保存	Secrets 扫描、运行时屏蔽、密钥轮换、禁止模型持有密钥

10.3 供应商与第三方模型

企业使用外部模型服务时，应明确数据是否用于训练、日志保留周期、区域与跨境、加密、隔离、访问审计、子处理方、SLA、漏洞通报和删除机制。对涉及高敏数据或高风险动作的用例，应采用更严格的部署形态和合约控制。

11. 评估、红队与安全指标

AI 应用的安全评估必须同时覆盖模型、提示模板、知识库、工具链、输出策略和业务流程。只测试基础模型的安全表现，不能代表整个应用安全。

11.1 评估集设计

- 正常样本：覆盖核心业务意图、语言、渠道、文件类型和用户角色。
- 攻击样本：直接注入、间接注入、越狱、数据外泄、系统提示泄露、敏感信息诱导、工具滥用。
- 边界样本：模糊请求、上下文不足、冲突证据、过期知识、低质量文档、跨语言输入。
- 回归样本：来自线上拒绝、误判、用户反馈、红队和安全事件的真实样本。

11.2 指标体系

指标	含义	风险解释
攻击成功率 ASR	攻击样本中突破防线的比例	越低越好；高风险场景应按攻击类型细分
敏感泄露率	输出或日志中出现敏感字段、密钥、越权内容的比例	必须接近零，且需要事件级追踪
工具误调用率	模型提出或系统执行了不应发生的工具调用比例	衡量 Agent 安全边界
拒答准确率	应拒绝的请求被拒绝、应回答的请求被回答的平衡	过严会影响体验，过松会放大风险
引用正确率	RAG 答案引用来源与结论匹配的比例	衡量事实可靠性和可追溯性
人工升级率	进入人工复核的比例	用于权衡风险与运营成本
安全延迟开销	安全网关、策略校验、复核带来的端到端延迟	用于架构容量规划
单位请求成本	模型、检索、工具和安全检查的综合成本	用于滥用检测和预算控制

11.3 红队方法

红队应采用“目标驱动”而非“花式越狱”导向。对企业而言，真正有价值的测试问题是：能否越权读取某类数据？能否诱导 Agent 执行高风险动作？能否让输出进入下游系统造成注入？能否通过知识库污染长期影响答案？能否在日志系统中泄露隐私？

每次红队结束后，应形成可复现攻击链、影响范围、根因、修复建议和回归样本。红队样本不应只作为报告附件，而应进入 CI/CD 安全评估流水线。

12. 运行监控、事件响应与持续治理

12.1 运行时遥测

AI 应用日志需要比传统应用日志更丰富，但也更容易泄露敏感信息。建议记录结构化元数据而非全量明文上下文，包括用户、租户、用例、模型版本、提示模板版本、检索文档 ID、工具调用、策略命中、拒绝原因、输出风险标签、成本和延迟。高敏原文可按需加密保存或不保存。

12.2 告警规则

- 同一用户短时间内大量触发系统提示泄露、越狱或敏感数据探测。
- 模型输出命中密钥、Token、PII、内部 URL、数据库表名或不应公开的专有名词。
- Agent 工具调用次数、失败重试、外部请求、成本或上下文长度异常升高。
- RAG 检索结果大量来自低可信来源、过期文档或用户上传文档。
- 某提示模板或模型版本上线后，拒答率、投诉率、敏感命中率或工具误调用率突变。

12.3 事件响应

阶段	关键动作
发现	确认告警、保全日志、识别用户、模型、提示版本、知识来源和工具链路
遏制	关闭高风险工具、降级模型能力、撤回知识库片段、提升人工复核、限制用户会话
根因分析	判断是提示模板、RAG 权限、输出校验、工具授权、供应商变更还是业务策略缺陷
修复	更新策略、补充校验、重建索引、回滚模型/提示、轮换密钥、更新权限
复盘	沉淀攻击样本、更新评估集、修订分级标准、补充监控指标和责任人

12.4 治理机制

建议建立 AI 应用安全评审委员会或等效机制，覆盖安全、数据、法务、合规、业务、研发和运维。治理重点不是阻止 AI 创新，而是把可接受风险、上线门禁、事故责任和持续改进机制明确化。ISO/IEC 42001 强调通过 AI 管理体系建立、实施、维护和持续改进组织内的 AI 管理过程[4]；这与企业 AI 应用安全治理高度一致。

附录 A：应用层控制清单

编号	控制项	验收标准
A1	AI 应用资产清单	所有模型、提示模板、知识库、工具、数据流和责任人已登记
A2	用例风险分级	每个用例标注数据敏感度、动作影响、用户范围和最低控制要求
A3	提示模板版本化	生产提示词具备版本、审批、灰度、回滚和变更记录
A4	上下文来源标注	系统指令、用户输入、检索内容、工具返回在结构上可区分
A5	RAG 检索前权限过滤	未授权文档不会进入候选召回集
A6	知识库入库审核	文档来源、敏感级别、版本、时效和所有权可追踪
A7	工具最小权限	AI 专用服务账号权限受限，并有速率、对象、金额和动作限制
A8	结构化输出校验	进入下游系统的输出必须通过 schema、策略和业务规则校验
A9	敏感信息检测	输入、上下文、输出、日志均有 PII、密钥和商业秘密检测/脱敏
A10	高风险人工确认	外部发送、资金、权限、删除、生产变更等动作需要确认
A11	红队评估	上线前与重大变更后执行攻击样本评估，结果进入门禁
A12	运行监控	具备异常成本、攻击意图、工具误调用、敏感输出和策略突变告警
A13	事件响应	有 AI 安全事件分级、遏制、复盘与回归样本沉淀流程
A14	供应商评估	外部模型和插件的数据使用、日志、隔离、合规、漏洞通报已评估
A15	审计证据	关键控制有可导出的日志、报告、审批记录和评估结果

附录 B：风险登记表示例

风险	触发条件	影响	现有控制	残余风险	负责人
间接提示词注入	AI 总结外部网页或邮件并可调用工具	数据外泄或误执行	上下文隔离、工具确认、输出网关	中	应用安全负责人
RAG 越权召回	用户查询与高敏文档语义相似	隐私/商业秘密泄露	检索前 ACL 过滤、租户隔离	低-中	数据平台负责人
输出下游注入	模型输出被前端或脚本解释执行	XSS/命令执行/数据破坏	编码、schema、沙箱	低	研发负责人
工具过度代理	Agent 自动串联多个高权限 API	资金、权限或生产系统误操作	Tool Broker、限额、人工审批	中	平台负责人
日志敏感泄露	Prompt/Response 全量入日志平台	二次数据泄露	脱敏、加密、访问审计、保留期限	低-中	安全运营负责人

附录 C：术语表

术语	定义

术语	定义
提示词注入	攻击者通过输入或间接内容诱导模型忽略或改变原有指令、泄露信息或执行非预期动作。
间接提示词注入	恶意指令隐藏在模型读取的外部内容中，例如网页、邮件、文档或工具返回值。
RAG	检索增强生成，通过外部知识检索为模型生成提供上下文证据。
Agent	具备规划、工具调用和多步执行能力的 AI 应用形态。
输出安全网关	在模型输出进入用户界面或业务系统前执行内容、格式、隐私和业务策略检查的组件。
攻击成功率 ASR	在给定攻击样本集中，攻击达到预定目标的比例。
模型防火墙	对模型请求和响应进行策略控制、检测和审计的运行时安全组件。
AIBOM	AI Bill of Materials，用于描述 AI 系统中模型、数据、组件、依赖和供应链信息的清单。

参考文献

[1] NIST. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile, NIST AI 600-1, 2024. https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf

[2] OWASP GenAI Security Project. OWASP Top 10 for LLMs and Generative AI Applications 2025. https://genai.owasp.org/llm-top-10/

[3] UK National Cyber Security Centre. Prompt injection is not SQL injection, 2025. https://www.ncsc.gov.uk/blog-post/prompt-injection-is-not-sql-injection

[4] ISO. ISO/IEC 42001:2023 Artificial intelligence — Management system. https://www.iso.org/standard/42001

[5] MITRE. ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems. https://atlas.mitre.org/

[6] Cloud Security Alliance. AI Controls Matrix, 2025. https://cloudsecurityalliance.org/artifacts/ai-controls-matrix

[7] Google. Secure AI Framework (SAIF). https://safety.google/intl/en_in/safety/saif/

[8] NIST AI Resource Center. AI RMF Playbook and trustworthy AI characteristics. https://airc.nist.gov/airmf-resources/playbook/