TC260-PG-2026NA

网络安全标准实践指南

——智能体系统开发安全指南

（征求意见稿 v1.0-202609）

全国网络安全标准化技术委员会秘书处

2026 年 09 月

本文档可从以下网址获得：
www.tc260.org.cn/

全国网络安全标准化技术委员会
National Technical Committee 260 on Cybersecurity of SAC

前　言

《网络安全标准实践指南》（以下简称《实践指南》）是全国网络安全标准化技术委员会（以下简称“网安标委”）秘书处组织制定和发布的标准相关技术文件，旨在围绕网络安全法律法规政策、标准、网络安全热点和事件等主题，宣传网络安全相关标准及知识，提供标准化实践指引。

本文件起草单位：浦江国家实验室、中国电子技术标准化研究院、中国移动通信集团有限公司、中国软件评测中心（工业和信息化部软件与集成电路促进中心）、上海阶跃星辰智能科技股份有限公司、中兴通讯股份有限公司、北京火山引擎科技有限公司、上海文鳐信息科技有限公司、安徽星盾智能科技有限公司、杭州网易智企科技有限公司、科大讯飞股份有限公司、蚂蚁科技集团股份有限公司。

本文件主要起草人：胡侠、孟令宇、周睿康、郑佳琪、郝春亮、张妍婷、乔兴格、李子威、徐阳、徐良、胡皓、唐刚、韩运宝、黄峥、李一冉、鲍景雨、夏云浩、郭建领、仲凯韬、潘宏博、王淼、黄爽、梅瑞、朱浩齐、苗晴晴、刘俊华、殷标、杨瑞、林冠辰。

声　明

本《实践指南》版权属于网安标委秘书处，未经秘书处书面授权，不得以任何方式抄袭、翻译《实践指南》的任何部分。凡转载或引用本《实践指南》的观点、数据，请注明“来源：全国网络安全标准化技术委员会秘书处”。

摘　要

智能体系统在自主决策、任务规划、工具调用、记忆管理和多智能体协作过程中，可能面临提示注入、目标偏离、权限滥用、数据泄露、记忆污染、供应链风险和级联失效等安全风险。本文件围绕软件智能体系统开发生命周期，提出准备、设计、实现、测试、发布阶段的安全开发指南，为开发者开展风险识别、安全控制设计、安全能力实现、安全验证和安全发布提供指引，可以为相关评估活动提供参考。

目　录

1 范围

本文件给出了基于大模型的智能体系统在准备、设计、实现、测试、发布阶段的安全开发指南。

本文件适用于开发者开展软件智能体系统开发活动，也可为相关评估活动提供参考。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 45654-2025 网络安全技术 生成式人工智能服务安全基本要求

3 术语定义

3.1 智能体系统 agentic AI system

由一个或多个智能体，以及支撑其运行的模型、工具、数据、外部服务、运行环境等组成，为完成特定任务而构建的系统。

注：本文件所称智能体系统主要指基于大模型构建的软件智能体系统，不包括以机器人、自动驾驶等物理设备控制为主要功能的智能体系统。

3.2 工具 tools

智能体用于获取信息、访问资源或者执行操作的外部能力，通过标准化接口供智能体调用，并定义相应的输入参数和输出结果。

3.3 技能　skills

智能体应用中封装了业务流程的可加载、可执行、可复用的功能单元。

3.4 上下文　context

在智能体运行过程中，影响其推理、规划、决策和执行的信息集合。

注：上下文可包括系统指令、开发者配置指令、用户输入、检索内容、工具返回结果、交互历史、任务状态和环境信息等。

3.5 模型上下文协议服务　model context protocol service

遵循模型上下文协议，为智能体提供工具、资源或提示等外部能力的服务。

3.6 记忆　memory

智能体应用针对用户构建的用于改进决策、适应性和性能的信息集合，可在任务内短期或跨任务长期存储和使用。

注：根据存储期限和使用范围，记忆可分为在当前任务或者会话中使用的短期记忆，以及跨任务或者跨会话持续保存的长期记忆。

3.7 自主执行　autonomous execution

智能体在无需人工逐步确认的情况下，根据设定目标、任务计划和约束条件，自主选择并实施操作的过程。

注：自主执行可包括任务分解、工具选择、工具调用、执行状态判断、异常处理以及根据执行结果调整后续操作等活动。

3.8 提示注入　prompt injection

攻击者通过构造恶意输入，将其注入智能体的提示或上下文中，使模型将恶意输入视为合法指令处理，从而绕过系统安全约束、覆盖

原有指令或执行非预期行为的攻击方式。

注：提示注入可分为直接提示注入（攻击者直接在用户输入中嵌入恶意指令）和间接提示注入（恶意指令隐藏于模型处理的外部数据中，如网页、文档或数据库记录等）。

3.9 安全护栏 security guardrail

在智能体系统运行过程中，为约束智能体的决策、规划和执行行为，确保其符合预期目标、安全策略和授权范围而设置的安全控制机制集合。

注：安全护栏包括输入检查、输出检测、提示注入防护、权限控制、行为约束、风险识别、安全策略触发、异常检测和处置等措施。

4 缩略语

API：应用程序编程接口（Application Programming Interface）

MCP：模型上下文协议（Model Context Protocol）

5 概述

5.1 适用对象

本文件适用于软件形态的智能体系统开发活动。智能体系统开发者可采用自主开发、平台化构建、低代码或零代码配置、系统集成等方式开展智能体系统开发活动，并根据其开发模式、技术实现方式以及可控制范围，结合本文件相应章节开展安全开发活动。

对于采用第三方提供的模型、开发框架、工具、插件、MCP 服务、API 接口、数据服务等组成要素构建智能体系统的开发者，开发者需结合其引入、配置、集成和使用情况落实相应安全要求。

5.2 开发原则

开发者可根据开发模式、技术实现方式以及可控制范围，结合本文件相应章节开展安全开发活动，并遵循以下原则：

a）风险导向原则：根据智能体的自主程度、应用场景及潜在影响范围，识别和评估安全风险，并实施差异化安全管理；

b）分级管控原则：根据智能体执行行为的操作风险等级，采取相匹配的控制措施，强化高风险操作约束和审查；

c）人类可控原则：根据智能体自主程度和任务风险设计人工监督与干预机制，明确职责边界，确保高风险行为可控制；

d）纵深防御原则：避免依赖单一安全控制措施，通过多层次安全控制、能力约束和风险隔离措施降低智能体系统整体安全风险；

e）全链路可追溯与可解释原则：对智能体关键行为、任务执行过程和关键决策依据进行记录、关联和分析，以支持行为追踪、决策解释、风险识别和责任分析；

f）持续优化原则：持续评估智能体系统安全风险，并根据系统能力、应用场景或运行环境变化，动态调整安全策略。

6 准备阶段

本部分提出智能体系统开发前的准备工作的安全指南，包括应用需求分析、安全需求分析以及风险识别等方面。

a）应用需求分析：明确智能体系统建设目的、应用场景、服务对象、业务任务、功能需求和预期能力，分析采用智能体系统实现相关功能的必要性和合理性。

b）自主能力与行为边界分析：根据业务需求、自主程度、数据敏感性和任务风险等因素，明确智能体自主运行范围、人工介入

要求以及行为约束边界，分析智能体可执行任务类型、可访问资源范围和禁止执行行为，为后续安全设计、权限控制和运行约束提供依据。

c）资产识别与攻击面分析：识别智能体系统涉及的数据、模型、工具、外部服务、用户接口等关键资产，分析数据流转过程、交互路径和潜在攻击面，为后续安全分析提供依据。

d）威胁分析：结合智能体系统的应用场景和技术架构识别智能体安全风险，智能体安全风险包括但不限于提示注入、上下文污染、记忆滥用、工具调用风险、自主决策风险以及多智能体协同风险等。风险识别维度包括智能体自主程度、工具调用权限、数据敏感程度、操作可逆性、外部影响范围和用户规模等维度。

e）安全需求分析：将安全分析结果形成风险登记册，记录风险来源、影响、责任人、控制措施、验证方式、剩余风险和接受或处置结论，并根据分析结果明确安全需求，包括权限控制、人工审批、运行隔离、日志审计、异常检测等，高风险操作清单和安全需求可作为设计、测试和发布门禁的输入。

f）安全合规分析：根据智能体系统应用场景、数据处理活动、服务对象和部署方式等因素，结合适用的法律法规和标准规范，进行安全合规分析，确保符合相关行业监管要求。

7 设计阶段

本部分提出智能体系统设计阶段的安全要求，将安全要求融入智

能体系统架构和能力设计过程，为后续工程实现、测试验证和安全管理提供基础。

a）模型安全选型：根据智能体系统业务目标、功能需求和安全需求开展模型选型，包括：

1) 采用符合GB/T 45654-2025要求的模型，并考虑模型能力与应用场景的匹配性；

2) 采用开源模型、自研模型或经过调整的模型时，分析可能引入的安全风险。

b）系统架构设计：开展智能体系统架构安全设计，包括：

1) 明确模型组件、智能体能力模块、数据资源、外部能力以及用户交互接口等系统组成要素之间的关系，形成合理的系统架构；

2) 根据系统组成、数据流转和功能调用关系划分安全边界和信任边界，明确不同组件之间的访问关系和交互约束；

3) 分析用户交互、模型调用、数据访问、工具调用和外部服务交互等关键流程中的数据流和调用关系，识别潜在攻击面和安全风险；

4) 根据风险分析结果设计相应安全控制要求，包括访问控制、输入校验、权限约束、风险隔离和异常处理等措施。

c）提示与指令安全设计：对提示与指令内容进行安全设计，包括：

1) 明确系统指令、开发者配置指令和用户输入等不同来源内容的安全属性和信任关系，确定不同类型指令之间的优先级和处理规则；

2) 根据智能体功能目标、安全要求和授权范围设计系统指令，明确智能体行为边界、任务约束和安全规则；

3) 设计关键指令和安全策略的保护要求，明确访问范围、修改权限和变更管理要求；

4) 针对提示注入、指令覆盖等风险设计防护策略，明确非可信指令识别、约束和处置要求。

d）上下文安全设计：设计智能体上下文管理机制，包括：

1) 识别用户输入、检索内容、工具返回结果、外部数据等上下文来源，明确其可信程度和使用规则；

2) 设计上下文来源标识、访问控制、内容校验和隔离控制等机制，降低非可信上下文信息影响智能体行为以及不同用户、不同租户之间上下文信息泄露或混用的风险；

3) 针对跨步骤、跨轮次、跨会话、跨用户、跨租户或跨智能体传递的上下文信息，设计可信性校验和隔离控制要求，防止未经授权的上下文信息传播或混用影响智能体行为；

4) 对知识库、检索索引及相关嵌入数据等上下文来源，设计完整性保护和安全管理措施，降低数据污染风险。

e）记忆安全设计：设计智能体记忆管理机制，包括：

1) 设计智能体记忆管理机制，明确短期记忆和长期记忆的管理要求；

2) 设计记忆访问控制和完整性保护机制，防止未经授权访问、修改或删除记忆内容；

3) 根据用户、租户、会话和任务等维度设计记忆隔离要求，明确不同类型记忆数据的访问范围、更新规则和生命周期要求；

4) 对长期记忆写入过程设计授权控制或策略审批要求，涉及个人信息或业务敏感信息时采取必要保护措施。

f）任务规划安全设计：设计智能体任务规划安全机制，包括：

1) 设计任务规划安全机制，明确用户授权意图、任务目标、执行范围和操作限制之间的对应关系；

2) 设计任务分解、执行流程和状态转换规则，明确不同任务阶段的安全约束要求；

3) 根据任务风险等级、操作影响范围和授权范围设计人工审批节点和执行限制要求，防止任务规划过程偏离用户意图或扩大授权范围。

g）自主执行安全设计：对智能体自主执行过程进行安全设计，包括：

1) 根据用户授权范围和任务风险等级确定智能体自主执行边界，对超出授权范围的操作进行限制或重新授权；

2) 结合智能体自主程度、任务复杂程度、工具调用权限、工具调用链复杂程度、数据敏感程度、操作可逆性和潜在影响范围等因素分析执行风险；

3) 根据风险情况设计必要的执行约束，包括最大执行步骤、模型调用次数、工具调用次数、执行时间、重试次数、资源消耗限额和外部服务调用额度等；

4) 设计自主执行异常检测和安全干预机制，明确异常循环、目标偏移、权限变化、资源消耗异常等风险识别和处置策略；

5) 对可能产生不可逆影响或具有外部副作用的操作，设计回滚、撤销或者补偿机制。

h）外部能力调用安全设计：设计智能体系统外部能力调用安全机制，包括：

1) 遵循最小权限要求设计外部能力管理机制，明确工具、技能、API接口、插件、MCP服务等外部能力的功能范围、适用场景、数据访问范围和调用权限要求；

2) 设计调用主体身份认证和权限控制机制，确保调用主体身份可信，并具备访问目标工具或资源所需的权限；

3) 设计外部能力调用授权机制，明确用户授权范围、智能体执行权限和外部资源访问权限，防止智能体利用已有访问权限执行未经授权的操作；

4) 根据外部能力调用风险等级设计调用限制、异常检测和审计要求，对高风险调用进行必要的约束和记录。

i）权限与人机协同设计：设计智能体系统权限关系与人机协同安全机制，包括：

1) 设计用户、智能体、工具服务和外部资源之间的身份关系和授权关系，明确身份认证、权限控制、授权范围管理和凭据生命周期管理要求；

2) 根据任务风险等级、自主执行能力和操作影响范围，明确需要人工参与的任务类型和适用场景；

3) 设计人工确认、人工复核、人工接管等干预机制，明确人工介入触发条件、处理流程和操作方式，确保人工能够在必要情况下了解任务状态并对智能体行为进行控制；

4) 对超出预设授权范围、涉及高风险操作或可能产生外部影响的行为，设计用户确认或人工审批机制；

5) 明确用户、智能体和人工操作人员之间的角色关系、权限范围和控制边界，设计高风险操作中的授权确认和行为记录要求，支持智能体行为追踪和责任分析。

j）多智能体协同安全设计：采用多智能体架构时，设计智能体协同安全机制，包括：

1) 明确不同智能体的角色、职责和权限范围，防止无约束任务转移；

2) 设计智能体之间身份关系、通信安全和权限控制机制，防止未授权智能体接入协同流程；

3) 设计任务委派和能力调用约束要求，防止权限扩散和越权执行；

4) 分析多智能体协同行为风险，并设计必要的隔离和控制措施。

k）数据安全设计：设计智能体系统数据安全保护机制，包括：

1) 根据数据类型和敏感程度设计数据保护策略，明确用户输入、上下文数据、记忆数据、工具调用数据等保护要求；

2) 设计数据访问控制、数据隔离和数据使用约束机制，防止未经授权访问或数据滥用；

3) 明确数据留存期限与销毁策略，防止数据超期存储和未授权复用；

4) 涉及个人信息或重要数据时，遵循最小必要原则，并设计相应的保护措施。

l）安全可观测设计：设计智能体系统安全可观测机制，包括：

1) 明确需要采集、关联和追踪的安全事件范围，包括用户请求、模型调用、上下文变化、任务执行、工具调用、权限操作和人工审批等信息；

2) 设计关键行为记录和关联分析要求，支持任务执行过程

追踪、异常分析和责任定位；

3) 针对任务规划、工具调用、安全策略触发等关键行为，记录必要的关联信息和决策依据，支持智能体行为分析和解释。

8 实现阶段

本部分提出智能体系统实现阶段的安全指南，主要依据第七章确定的安全架构和安全要求开展实现，同时结合智能体系统实现过程中的工程实践要求，提出安全编码、供应链管理、运行环境配置等安全措施。

a）系统架构与安全能力实现：根据设计要求开展智能体系统架构与安全能力实现，包括：

1) 实现模型、数据、工具、外部服务和用户交互接口等组件之间的安全连接与交互控制；

2) 实现组件之间的数据传输、接口调用和状态交互机制，确保组件连接关系符合设计要求；

3) 实现系统边界、信任边界和访问控制要求，防止组件之间发生未经授权的数据访问或能力调用；

4) 对系统组件配置、接口连接关系和关键依赖进行版本管理，支持系统变更控制。

b）提示与指令安全实现：根据设计要求实现提示与指令安全，包括：

1) 将智能体行为约束、安全策略和任务限制要求配置至系统指令或相关安全控制模块中；

2) 对系统指令、开发者配置指令和用户输入等不同来源内容进行区分管理，防止低可信来源内容覆盖安全约束；

3) 实现提示注入、指令覆盖、恶意输入等风险防护措施，降低非可信指令影响智能体行为的风险；

4) 对系统指令、安全策略和提示模板等关键配置进行访问控制、版本管理和变更审计。

c）上下文安全实现：应实现上下文安全控制，包括：

1) 实现用户输入、检索内容、工具返回结果和外部数据等上下文来源识别和分类管理能力；

2) 实现上下文可信校验、来源标识和隔离控制机制，降低提示注入、上下文污染以及不同用户、不同租户之间上下文混用风险；

3) 对知识库、检索索引、嵌入数据等影响智能体推理结果的重要数据资源实施完整性保护和变更管理，防止未经授权修改或数据污染；

4) 对提示模板、安全策略和记忆配置等关键内容进行统一管理，支持安全更新和版本控制。

d）记忆安全实现：应实现智能体记忆安全管理能力，包括：

1) 按照设计要求实现短期记忆和长期记忆隔离，防止跨用

户、跨租户和跨会话访问；

2) 实现记忆访问控制、更新控制和删除机制，防止未经授权读取、修改或删除记忆内容；

3) 对长期记忆写入过程进行安全控制，防止恶意内容注入和错误信息长期保存；

4) 对记忆数据删除后的副本、缓存和索引进行清理或失效处理。

e）规划与自主执行安全实现：应实现智能体运行控制安全机制，可通过系统原生能力或集成外部安全能力实现，包括：

1) 实现任务分解、执行流程和状态管理机制，确保任务执行符合设计目标和授权范围；

2) 实现执行步骤、模型调用次数、工具调用次数、执行时间、重试次数、资源使用量和外部服务调用额度等限制措施；

3) 实现异常循环、目标偏移、权限变化和资源异常消耗等风险检测能力，并采取暂停、终止、限制执行或转人工处理措施；

4) 对不可逆操作或具有外部副作用的操作，实现人工确认、回滚、撤销或补偿机制，并记录相关信息；

5) 实现用户授权范围管理机制，确保智能体执行任务过程中持续遵循授权边界。

f）外部调用能力安全实现：应实现外部能力调用安全机制，包

括：

1) 对高风险调用实施频率限制、异常检测和调用审计；

2) 结合用户授权、任务目标、数据敏感程度和调用场景等因素，对智能体工具调用行为进行动态调整，防止未经授权、超出业务目的或高风险的工具调用；

3) 对外部能力输入参数和返回结果进行安全检查，防止恶意输入、异常结果或非可信信息影响智能体行为；

4) 完整记录外部能力调用的主体、时间、输入参数、返回结果等核心信息，支撑安全审计和问题追溯；

5) 对外部能力的来源进行真实性验证和完整性校验，防止接入恶意或被篡改的外部实体；

6) 实现调用前授权校验机制，对工具调用行为进行授权范围验证，防止超出用户授权范围调用外部能力。

g）身份认证与授权管理实现：实现智能体系统身份认证和授权管理能力，包括：

1) 对用户、智能体、工具服务、外部服务等交互主体进行身份识别和认证，确保访问主体身份可信；

2) 实现智能体授权关系管理机制，对用户授权范围、智能体执行权限和外部资源访问权限进行关联管理，确保智能体代表用户执行任务时符合授权范围；

3) 按照最小权限原则配置智能体及相关组件的访问权限，

限制对模型、工具、数据和外部服务的访问范围；

4) 对密钥、访问令牌、证书等敏感凭据进行安全存储、使用和更新，防止泄露、滥用或未经授权访问；

5) 对身份认证、授权决策和凭据使用过程进行记录，支持安全审计和问题追踪。

h）人机协同安全实现：实现智能体人机协同安全机制，包括

1) 实现人工确认、人工复核和人工接管机制，根据设计要求支持高风险任务暂停、审批、终止或恢复执行；

2) 实现人工干预过程的状态展示能力，使操作人员能够了解任务执行状态、关键操作信息和风险提示；

3) 对人工操作过程实施身份认证增强、操作日志记录和异常行为检测，防止账号被盗用或内部人员滥用权限；

4) 记录人工干预过程中的操作主体、操作时间、操作内容和处理结果等信息，支持安全审计和问题追踪。

i）多智能体协同安全实现：实现多智能体协同过程中的安全控制能力，包括：

1) 实现智能体身份识别和认证机制，确保参与协同的智能体身份可信，防止未授权智能体接入协同流程；

2) 根据设计要求实现智能体间通信保护机制，对协同消息进行必要的完整性保护和安全校验，防止消息篡改、伪造或非可信信息传播；

3) 实现任务委派和权限控制机制，对智能体之间的任务分配、能力调用和权限传递进行约束，防止权限扩散和越权执行；

4) 实现多智能体协同行为监测和异常处置机制，对异常通信、异常调用、任务偏移等风险进行检测，并采取限制、隔离或终止措施。

j）数据安全实现：根据设计要求落实数据访问控制、敏感信息保护、数据生命周期管理等措施，防止数据未经授权访问、泄露或滥用。

k）安全编码实现：开展安全编码活动，包括：

1) 对代码执行、工具调用、外部接口访问等涉及外部交互的功能进行安全控制，防止未经授权的操作、命令执行或资源访问；

2) 对输入参数、接口请求和工具调用结果进行安全校验，防止恶意输入、异常数据或非可信返回结果影响智能体运行行为；

3) 实施必要的权限控制和异常处理机制，防止因代码缺陷导致越权访问、权限绕过或安全控制失效；

4) 避免在代码、配置文件、提示词、上下文数据或者普通日志中明文存储密钥、令牌等敏感凭据；

5) 对关键代码变更开展安全检查，降低代码修改对智能体

行为和安全策略产生的不利影响。

l）安全护栏实现：构建智能体系统安全护栏，包括：

1) 根据业务场景、智能体任务目标和安全要求建立安全策略和风险规则库，明确输入内容、输出结果、任务执行过程和工具调用行为等安全控制要求；

2) 对高风险输入、高风险输出、异常规划行为、异常工具调用和越权执行行为采取拦截、告警、限流、暂停或终止执行等差异化管控措施；

3) 记录安全护栏触发情况，为风险分析和问题追踪提供依据，并宜根据系统运行情况持续优化安全策略。

m）供应链安全管理：开展供应链安全管理，包括：

1) 识别基础模型、智能体开发框架、第三方组件、工具、MCP服务、API接口和外部数据服务等供应链要素，记录其来源、版本、许可、维护状态和安全风险；

2) 对引入的第三方组件、依赖库和外部服务开展来源校验、完整性验证和版本管理，降低供应链投毒、组件漏洞和服务不可用风险；

3) 建立供应链漏洞跟踪和处置机制，及时识别、评估和处理已知安全漏洞；

4) 形成软件物料清单和人工智能相关组件清单，记录模型、框架、组件、插件、工具及外部服务等关键要素信息；

5) 对基础模型、智能体框架和外部能力服务的版本变化进行安全影响评估，防止变更引入新的安全风险。

n）运行环境安全管理：应实现运行环境安全控制能力，包括：

1) 将模型推理、工具执行、代码执行等执行活动置于适当的沙箱、容器或其他隔离边界内；

2) 默认限制不必要的网络访问、文件系统访问和系统调用权限；

3) 实施最小操作系统权限、资源配额、超时、并发及网络目的地控制，并对隔离策略失效、越权访问等异常行为进行检测，并采取告警、阻断或终止措施；

4) 禁止智能体直接访问不必要的敏感资源。

o）安全可观测与应急能力实现：实现智能体系统安全可观测和应急处置能力，包括：

1) 实现关键行为记录和关联分析能力，对用户请求、模型调用、上下文变化、任务执行、工具调用、权限操作、人工审批等关键活动进行记录，并采取日志完整性保护措施，防止日志被篡改或删除；

2) 实现任务执行过程关联记录能力，对任务规划、工具调用、安全策略触发等关键行为记录必要关联信息，为异常分析、行为追踪和责任分析提供支撑；

3) 实现异常检测和风险告警能力，对异常行为、安全策略

触发、权限异常和执行异常等情况进行识别，并支持风险告警和安全策略调整；

4) 实现安全事件处置能力，根据风险情况采取暂停、隔离、终止、回滚等措施，降低异常行为造成的影响。

9 测试阶段

本部分提出智能体系统测试阶段的安全指南，开发者可通过功能测试、安全测试和红队测试等测试方式，验证智能体系统在正常使用和对抗环境下的安全性。

a）用户交互安全测试：开展用户交互与输入安全测试，包括：

1) 测试提示注入、指令覆盖、越权诱导、语义绕过等攻击场景，验证智能体是否能够保持既定行为约束；

2) 测试多轮交互过程中恶意信息累积、上下文污染对智能体行为的影响；

3) 测试用户输入校验、输出过滤和内容安全控制机制；

4) 测试智能体对高风险请求的识别、拒绝和安全响应能力。

b）上下文安全测试：开展上下文安全测试，包括：

1) 测试外部数据、检索结果、工具返回信息对模型行为的影响；

2) 测试非可信上下文是否能够突破或覆盖系统指令或安全约束；

3) 验证上下文隔离、可信来源控制等机制有效性。

c）记忆安全测试：开展记忆安全测试，包括：

1) 测试短期记忆是否存在跨用户、跨会话泄露风险；

2) 测试长期记忆污染、错误信息注入和非法修改风险；

3) 测试记忆访问权限控制；

4) 验证记忆删除和生命周期管理机制。

d）外部能力调用安全测试：开展外部能力安全测试，包括：

1) 测试外部能力未授权调用、越权调用和权限绕过风险；

2) 测试外部能力参数注入、异常参数和恶意返回结果影响；

3) 测试外部能力调用链中的风险传递、权限扩散和异常调用风险；

4) 验证外部能力调用记录、异常终止和恢复机制的有效性。

e）规划与自主执行安全测试：开展任务规划与自主执行安全测试，包括：

1) 测试复杂任务、多步骤任务执行过程中的行为稳定性；

2) 测试目标偏移、异常循环、资源滥用等风险；

3) 验证任务限制、人工审批、暂停和终止机制；

4) 对任务执行轨迹进行分析，验证关键行为符合预期。

f）身份、授权与人机协同测试：开展身份、授权和人机协同安

全测试，包括：

1) 测试用户、智能体、工具和外部服务之间身份认证机制；

2) 测试权限提升、权限绕过和跨主体访问风险；

3) 测试高风险操作是否触发人工确认、人工复核或人工接管；

4) 测试用户授权范围是否能够正确约束智能体任务执行；

5) 测试智能体是否存在超出授权范围调用工具、访问数据或执行操作的风险；

6) 测试高风险操作重新授权、人工确认机制有效性。

g）多智能体协同测试：采用多智能体架构时，开展多智能体协同安全测试，包括：

1) 测试智能体身份认证、通信保护和消息完整性；

2) 测试恶意智能体接入、异常任务委派和权限扩散风险；

3) 测试多智能体异常通信、级联调用和风险隔离能力。

h）安全可观测与应急能力测试：开展智能体安全可观测能力与应急能力测试，包括：

1) 验证日志记录、行为追踪和审计能力；

2) 测试异常检测、风险告警和安全策略调整能力；

3) 测试熔断、隔离、恢复和回滚能力；

4) 验证安全事件相关日志与证据的完整性、防篡改性与留存合规性。

i）模型调用与模型安全测试：开展模型调用与模型安全测试，包括：

1） 测试模型能力边界和安全约束遵循情况，验证模型调用过程是否存在超出系统授权范围或安全要求的行为；

2） 测试模型输出可靠性，识别错误输出、有害输出和敏感信息泄露风险；

3） 测试模型版本变化、参数调整对系统安全性的影响。

j）合规性测试：开展智能体系统合规性测试，包括：

1） 根据智能体系统应用场景、服务对象和业务范围，检查系统涉及的数据处理、用户交互、模型调用、外部能力调用等活动是否符合相关法律法规和标准规范要求；

2） 检查智能体系统安全设计、安全控制措施和管理机制是否满足相关标准规范及组织安全要求；

3） 检查用户授权、身份认证、权限控制、数据保护、安全记录和审计追踪等机制的有效性；

4） 对发现的不符合项进行风险评估，并采取整改、优化或限制使用等措施。

k）测试过程管理：对智能体测试过程进行管理，包括：

1） 结合威胁分析和风险评估结果制定测试计划，覆盖正常、异常、滥用、对抗和故障等场景；当模型、提示、策略、工具、数据或依赖等关键组件发生变化时，开展回归测试；

2) 对测试环境和生产环境进行隔离，并对测试数据实施必要的访问控制、敏感信息保护和清理措施；

3) 记录测试结果，包含测试环境、系统版本、模型或服务版本、提示和数据版本、测试场景、测试输入、预期结果、实际结果、问题分析和复测结论等信息。

4) 对发现的问题进行风险评估，对高风险问题完成修复和重新验证，记录问题处置情况和剩余风险情况，并作为发布阶段安全评估和发布决策的依据。

10 发布阶段

本部分提出智能体系统发布阶段的安全指南，包括智能体系统上线前安全确认、发布过程安全控制以及上线后安全状态验证与记录留存等方面。

a）发布前安全评估：开展发布前安全评估，确认智能体系统满足上线要求，包括确认系统已完成安全测试，发现的安全问题已完成整改或已采取等效风险控制措施，并确认模型版本、提示模板、工具配置、安全策略等关键内容与测试版本保持一致。

b）发布前安全能力验证：确认发布版本已启用必要的安全防护能力，包括输入安全控制、工具调用约束、权限管理、日志审计、异常检测和人工干预等机制，确保发布版本具备设计要求的安全能力。

c）发布版本管理：建立发布版本管理机制，记录发布版本涉及

的系统组件、基础模型、提示模板、工具配置、安全策略等关键信息，确保发布内容可追踪，并支持必要情况下的版本回退。

d）发布过程控制：限制发布操作权限，确保发布过程经过授权，涉及基础模型替换、工具权限调整、安全策略变更等重大变化时，应重新开展安全评估与验证。

e）安全发布策略：采用适当的发布策略降低发布风险，高风险系统采用可观测的灰度或分阶段发布，并在发布过程中验证系统功能和安全控制措施有效性。

f）发布异常处置：建立发布异常处置机制，在发布过程中发现安全异常或系统异常时，宜采取暂停发布、版本回退等措施。

g）发布后安全能力验证：应对发布后的智能体进行安全能力验证，包括输入安全控制、工具调用约束、权限管理、日志审计、异常检测和人工干预等机制，确认智能体安全能力持续有效。

h）发布记录留存：建立发布记录和安全资料留存机制，记录发布版本、发布时间、发布内容、模型版本、工具配置、安全策略变更等关键信息，并保存相关安全评估、测试结果和问题处置记录，支持后续问题定位和安全追溯。

参考文献

[1] 人工智能安全治理框架 2.0

[2] 网络安全标准实践指南——智能体部署使用安全指引

[3] GB/T 45654-2025 网络安全技术 生成式人工智能服务安全基本要求

[4] GB/T 47470-2026 网络安全技术 软件安全开发能力评估准则

[5] ISO/IEC 22989:2022 Information technology-Artificial intelligence-Artificial intelligence concepts and terminology