

Permission Lifecycle: Grant, Lease, Revoke, Decay

权限生命周期治理——Agent 的权限什么时候该消失

2026年10月25日 | AI Agent 知识分享 | 第85期

一、一句话概括

授权只是起点，权限什么时候该消失才是分水岭——Agent 的权限必须有完整的生命周期：按需申请、限时持有、到期自动衰减、撤销立即生效、事后可审计。第 46 期讲的是 Agent 怎么拿到权限、怎么保管凭证，这一期讲的是另一半：权限什么时候该收回来。只做授权不做回收的系统，三个月后安全审计会查出一堆没人认领的僵尸权限。

二、核心原理

>> 1. 权限生命周期的五个阶段

申请 (Request)：任务开始前按需申请，不做预授权，不搞「先给一堆权限慢慢用」。授予 (Grant)：审批通过后下发带 TTL 的凭证，凭证上写清范围、时效、可续期次数。持有 (Hold)：执行期间有效，超出凭证声明的范围立即失效，不做「宽松解释」。衰减 (Decay)：随时间、闲置时长、使用次数或使用异常自动降级，从「可写」退到「只读」，从「只读」退到「无」。撤销 (Revoke)：任务结束、任务异常、人员变动、凭证疑似泄露时立即回收，不等下一次人工巡检。

>> 2. 四种权限时效模型（按风险从低到高）

会话级 (Session-scoped)：整个会话有效，会话结束即回收，适合交互式短任务。任务级 (Task-scoped)：单个任务内有效，任务进入终态（完成/失败/取消）即回收，适合后台批处理。凭证级 (Credential-scoped)：权限绑定具体 Agent 凭证，凭证一轮换权限立即失效，适合多 Agent 协作。即时级 (Just-in-Time)：单次调用有效，用完即焚，适合删除、转账、改权限这类高危操作。

>> 3. 撤销传播问题——撤销不是改一行状态

权限撤销本质上是一次分布式传播，至少有三个副本要同步失效：授权服务器的本地缓存、Agent 运行时的内存缓存、下游工具侧的缓存。传播延迟 = 从「撤销指令已发出」到「所有副本都失效」的时间差，在这段时间窗口里 Agent 仍然可以合法地越权调用。高危操作（删除数据、发起支付、修改权限）必须走同步撤销——撤销请求阻塞，直到确认所有副本失效才返回；低危只读操作可以异步撤销，但必须声明最大残留窗口。

>> 4. 权限衰减——长期持有权限等于长期暴露面

衰减不是「一刀切过期」，而是按行为证据降级：闲置超时降级、连续失败降级、异常行为立即收回全部权限。银行场景最典型：测试账号的临时授权、外包人员的限时权限、运维窗口期内的高权限，这三类都必须有自动衰减机制，不能靠人记得回收。

>> 5. 最小权限的实现不是「少给」，而是「短给」

静态最小权限：一次给足但只给必要的动作，问题是给的这段时间里权限一直在。动态最小权限：用时申请、用完即还，

这才是银行「最小授权、全程留痕」监管要求的落地形态。判断标准很简单：如果这个在凌晨三点被诱导，它当时手里的权限能不能把库删掉？能，就是给多了或者给长了。

Agent

核心要点

授权只是起点，回收才是终点——权限必须有 TTL，任务进终态必须触发回收钩子

撤销是一次分布式传播，至少有授权服务器缓存、Agent 内存缓存、下游工具缓存三个副本要同步失效

传播延迟 = 撤销指令发出 到 所有副本失效 的时间差，这段时间里 Agent 仍能合法越权

高危操作走同步撤销（阻塞确认），低危只读走异步撤销（声明最大残留窗口）

TTL 按风险分档：只读 1 小时、写入 15 分钟、危险操作单次有效；基于任务耗时 P95 滚动调优

僵尸权限是审计重灾区：离职人员的 Agent 服务账号、失败任务未回收的权限、已下线工具的权限

三、测试实战

◆ 测试一：权限 TTL 到期自动失效测试

目的：验证权限不会提前失效（误杀正常任务），也不会延迟失效（留下窗口）。

操作：给 Agent 下发 TTL=60s 的数据库查询权限。t=0s 执行查询（应成功），t=45s 再执行（应成功），t=65s 再执行（应被拒绝），重复 20 轮。同时验证续期：t=50s 任务未完成时能否自动续期，续期后 t=115s 是否仍成功。

工具：pytest (<https://pytest.org>) + 自定义 TimeProvider（把系统时钟抽象成可注入的 fake clock，不要真的 sleep 65 秒）+ WireMock (<https://wiremock.org>) 模拟授权服务器。

指标：到期前调用成功率 = 100%、到期后拒绝率 = 100%、边界点 (TTL±1s) 判定准确率 = 100%、假过期率 = 0。

◆ 测试二：撤销传播延迟测量

目的：量出「撤销指令发出」到「所有副本失效」的真实时间差，这个差值就是残留风险窗口。

操作：Agent 持有下游系统权限，t=0 触发撤销，之后每隔 100ms 探测一次能否继续调用该工具，记录最后一次成功调用的时刻。三种场景分别测：(a) 同步撤销（阻塞至所有副本失效）(b) 异步撤销（先发确认再传播）(c) 带缓存撤销（下游缓存 TTL=30s）。

工具：locust (<https://locust.io>) 做高频探测 + Redis 做权限缓存（验证缓存失效时间）+ OpenTelemetry (<https://opentelemetry.io>) 记录撤销事件时间戳。

指标：同步撤销传播延迟 < 50ms、异步撤销最大残留窗口 < 10s、带缓存场景残留窗口 = 缓存 TTL + 传播延迟（不能超出）、撤销后残留成功调用次数 = 0。

◆ 测试三：越权残留扫描（Orphaned Permission Scan）

目的：找出所有指向失效主体的权限条目——审计时最容易出事的一类。

操作：模拟三类残留场景：(a) 员工离职后，他创建的 Agent 服务账号仍持有权限 (b) Agent 任务失败退出但权限未回收 (c) 权限被授予给一个已下线的工具。写扫描器：从权限授予日志出发，交叉比对「在职人员表」「活跃 Agent 表」「活跃工具注册表」，沿「人→Agent→凭证→工具」四层关系找断链。

工具：Python 扫描脚本 + Neo4j 图数据库 (<https://neo4j.com>) 做权限关系图谱遍历 + 规则引擎比对。

指标：残留权限检出率 > 99%、误报率 < 1%、扫描覆盖率 = 100%、单轮全量扫描耗时 < 5 分钟。

◆ 测试四：权限衰减策略验证与误伤率

目的：在「误伤正常任务」和「及时限制异常行为」之间找到平衡点。

操作：三种衰减策略各跑 50 次任务对比：(a) 固定 TTL (60s 强制过期) (b) 闲置衰减（闲置超 10 分钟自动降级为只读）(c) 异常触发（连续 3 次调用失败立即收回全部权限）。对比正常任务完成率、权限被提前收回次数（误伤）、异常行为被及时限制的比例。

工具：hypothesis 做基于属性的测试 (<https://hypothesis.readthedocs.io>) + toxiproxy 做故障注入 (<https://github.com/Shopify/toxiproxy>)。

指标：误伤率 < 2%、异常限制及时率 > 95%、固定 TTL 场景下因权限到期导致的任务中断率 < 5%（超过说明 TTL 给短了）。

◆ 测试五：权限审计完整性与不可篡改性验证

目的：保证审计日志能作为合规证据——完整、有序、不可改。

操作：跑 100 个任务，每个任务涉及 5-10 次权限事件（申请/授予/使用/续期/撤销），用「预期事件清单」与「实际日志」做 diff，找出漏记、错记、时间戳乱序三类问题。特别检查：撤销事件是否记录了传播结果（成功/失败/超时）。

工具：OpenTelemetry (<https://opentelemetry.io>) 采集 + pytest 断言 + Allure 生成审计报告 (<https://github.com/allure-framework>)。

指标：审计事件完整率 = 100%、时间戳单调性 = 100%（不允许撤销时间早于授予时间）、撤销结果记录率 = 100%、日志哈希链校验通过。

四、常见坑点

坑点一：权限发出去就不管——授予容易回收难

最常见的坑是上线时只实现了授权，没实现回收。Agent

完成任务后权限一直挂着，三个月后安全审计发现几十个僵尸权限，谁也说不清是干什么用的。对策：把权限回收写进任务状态机——任务进入终态（完成/失败/取消）时必须触发回收钩子，回收失败要告警并进入重试队列。回收不是可选项，是任务结束的必要条件。

坑点二：撤销只改了主表，缓存还在放行

撤销请求只更新了授权数据库，但 Agent 运行时的内存缓存和下游工具的本地缓存还在按旧权限放行。管理员以为已经回收了，实际上攻击者或误操作还能再用 30 秒。对策：撤销必须走「先失效缓存、再确认返回」的顺序；所有权限缓存必须有 TTL，且 TTL 必须小于可接受残留窗口；对高危权限（删除、转账、改权限）直接禁用缓存，每次实时查询。

坑点三：TTL 设置拍脑袋——太长是风险，太短是事故

权限 TTL 设成 7 天，等于给了攻击者 7 天的窗口；设成 30 秒，正常任务跑到一半权限就没了，用户看到的是「系统又抽风了」。对策：TTL 按权限风险等级分档（只读 1 小时、写入 15 分钟、危险操作单次有效），并基于生产数据滚动调优——统计任务实际耗时分布，让 TTL 覆盖 P95 任务时长。同时必须支持自动续期，避免长任务被中途掐断，但续期要有次数上限和总时长上限，不能无限续。

五、延伸阅读

- OWASP API Security Top 10 (API 对象级授权失效 / API5 功能级授权失效) : <https://owasp.org/API-Security/editions/2023/en/0x11-t10/>
- NIST SP 800-207 Zero Trust Architecture (即时授权与最小权限官方框架) : <https://csrc.nist.gov/publications/detail/sp/800-207/final>
- OAuth 2.0 Token Revocation RFC 7009 (令牌撤销标准协议) : <https://datatracker.ietf.org/doc/html/rfc7009>
- HashiCorp Vault Dynamic Secrets (动态密钥与租约机制实现参考) : <https://developer.hashicorp.com/vault/docs/secrets>
- OpenTelemetry GenAI Semantic Conventions (Agent 权限事件采集与追踪规范) : <https://opentelemetry.io/docs/specs/semconv/gen-ai/>