“铸网2025”暨“磐石行动”2025年上海市
工业和信息化领域网络安全实战攻防活动总结大会

基于AI驱动的实战网络攻击

—— 更真实地模拟黑客以攻促防

吴飞飞（Feei） 首席网络安全官

支付宝支付科技有限公司

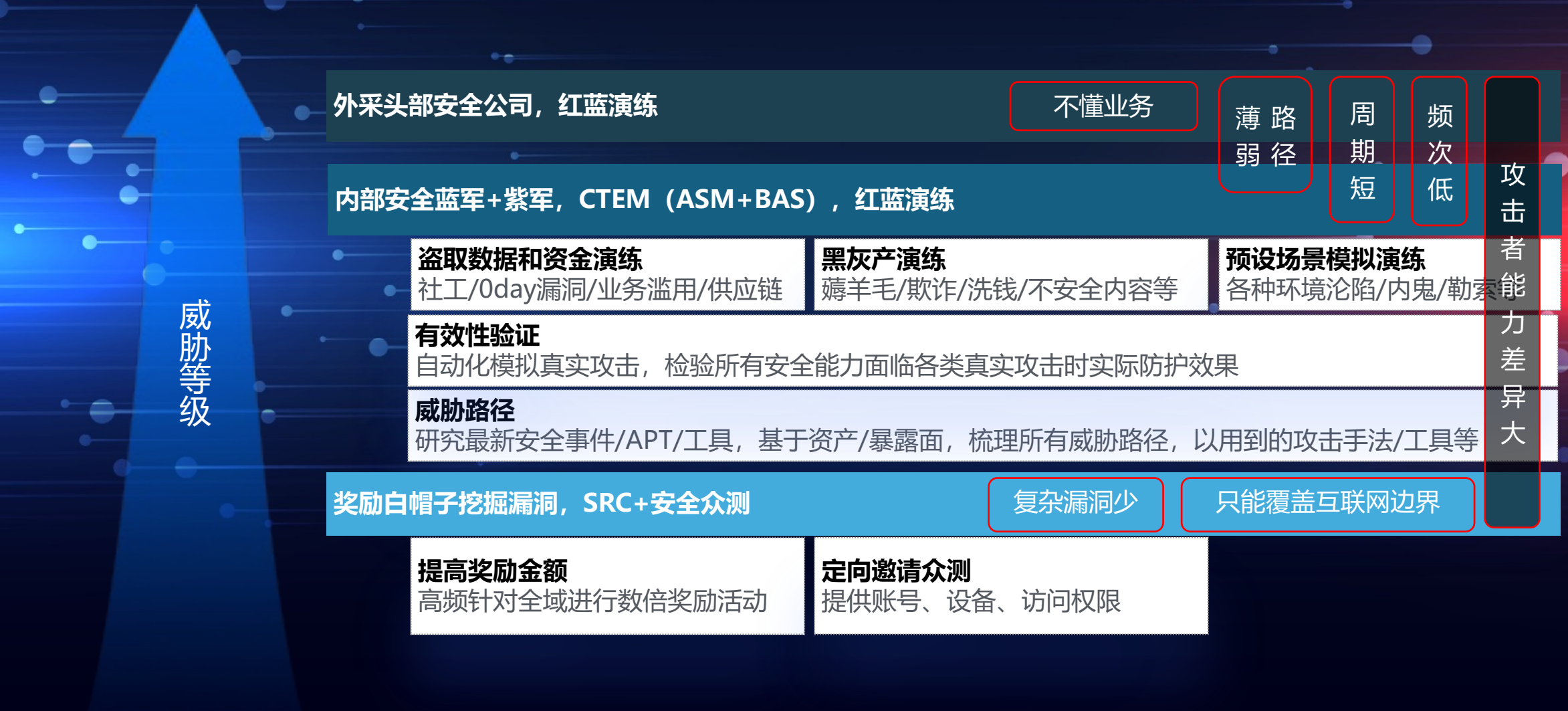
2025/11

目 录

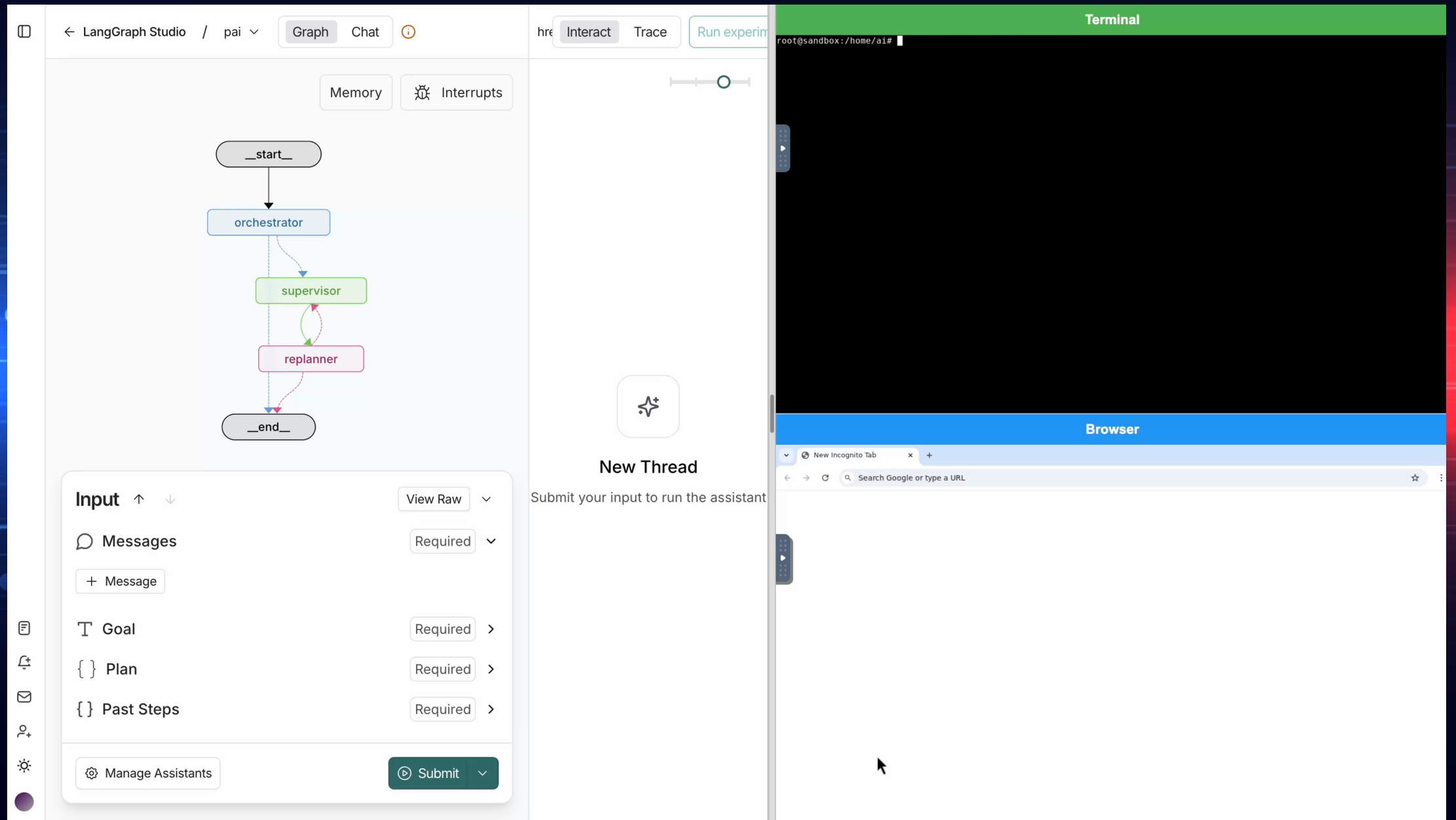
CONTENTS

当前实战网络攻击的瓶颈

期望目的：全面、深入、持续不间断实战检验，不断揭示新的薄弱点以及验证已知路径防护效果，降低被真实攻击者利用的风险。
理想状态：有一批各种背景各地的顶尖攻击者，他们非常了解支付宝，7x24小时不间断，对所有可能的入口，尝试了所有的攻击方法，并不断渗透扩大影响，拿到了大批量数据或资金。



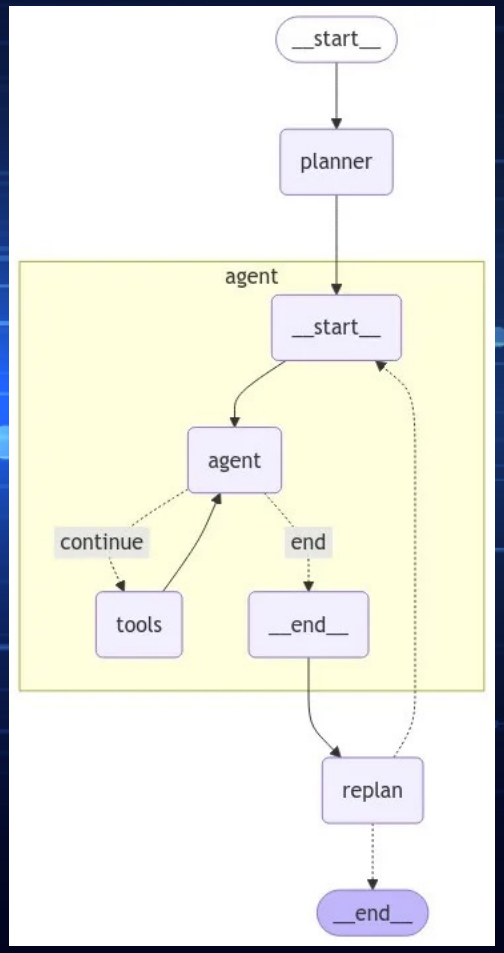
基于LangGraph+Multi Agent Demo



AI原生安全领域知识与能力瓶颈

初版Demo显示AI具备初级攻击者的能力，但离高级攻击者还有很远距离

能够根据用户指令理解攻击意图，拆解步骤，控制浏览器访问目标网站，理解网站功能，猜测并尝试各种可能的漏洞，最终发现SSTI漏洞，调用命令行监听端口实现反弹Shell。



LangGraph中基于DAG描述工作流

主Agent的规划基于

1、模型内生安全知识
2、预先提示词设定（比如对不同目标提前写的测试思路）
3、用户下发任务时指定的目的（比如要求测试网站的SSRF漏洞）

那么，决定安全攻击能力的因素为

1、基座大模型的能力
2、工程设计（做好多Agent协作，上下文管理，提供对应渗透工具等）
3、上下文素材（即各种漏洞挖掘以及渗透思路的提示词等）

即使做好上面三点，实际效果还是不理想，因为以上的设计都是把Agent变的足够通用，能够应对所有渗透测试场景，这会导致信息过载和测试目标和思路不明确，追求通用而忽视了专家经验。

比如，在对企业内部各种办公站点测试时，安全专家已经有一套测试模式。BUC->功能->可能风险尝试。这时候专家不会重复测试每个BUC，也不会用CVE去扫网站，但AI Agent会。

AI有原生安全领域知识和能力，但不多，如何将人类专家知识经验和能力融合起来？

AI特定领域场景应用的演变路径

AI工程正经历从面向模型到面向意图的深刻变革,每一代演进都在解决如何让AI更智能、更可控的核心问题。

第一代 - 面向基模 -> Prompt Engineering

通过精心设计提示词优化LLM输出，让AI更好地理解和响应人类的自然语言指令。
安全典型应用：询问某段代码可能的漏洞；生成钓鱼邮件等

1

第二代 - 面向Agent -> Context Engineering

基于ReAct、Plan-and-Execute等Agent设计模式，让AI更听话、更聪明、更准确，能够执行更复杂的任务。代表：Anthropic的Claude Code；
安全典型应用：安全RAG知识库沉淀与答疑；通过MCP/Tool工具与现有安全系统互动；

2

第三代（当下） - 面向意图 -> Intent Engineering

通过意图工程实现将专家的意图精确表达和执行
GitHub Speckit；蚂蚁集团 HOP

3

意图的理解

如何让AI更好地理解我们的意图。开源代表:Speckit。通过更精确的意图表达方式,使AI能够准确把握用户的真实需求。

意图的执行

如何让AI坚决执行我们的意图。开源代表:HOP。确保AI在理解意图后能够按照预期的方式执行,不偏离轨道。

基于AI驱动的实战网络攻击 – 设计思路

【攻击知识】APG（Attack Pattern Graph，攻击模式图），用图结合自然语言沉淀专家渗透测试经验

APG是专家渗透测试经验的沉淀和表现形式，使用图的节点和边以及属性能够很形象描述渗透测试过程。本质是一种高维 Prompt，其中包含对Agent行为的指导，工具的使用建议，上下文的加载策略等，这些元素都可以作为节点和边的属性。虽然元素很多，但是不需要用复杂的代码去表达，只需要简单的自然语言即可，实现从文档化知识到可执行知识。

【理解执行】APG Runtime，理解并执行APG

APG Runtime主要由解释器和执行引擎组成，沿着APG的节点路径移动执行，负责图遍历、Prompt编译、上下文管理和工具调度。维护当前执行状态，根据节点输出和边条件决定下一步转移。将节点定义和上下文编译为可执行的AI提示，驱动LLM执行。可以避免因为大模型幻觉问题导致每次执行结果不一致问题。

【攻击工具】Offensive Infrastructure，成熟安全工具的AI使用友好化及AI能力强化

提供AI Native的基础工具集（操作浏览器/命令行及各类App等）以及攻击工具集（包括信息收集/漏洞利用/权限维持等完整工具链），让AI不用每次低效的去尝试攻击（比如爆破/反弹Shell等），而是使用现成的工具去提高攻击效率。同时，让AI能力赋能部分工具的能力，比如钓鱼邮件生成、环境安全检测等。

观测指挥平台

人类只需要提供目标，之后的意图理解、任务拆解、信息收集、漏洞利用、权限维持、横向渗透、数据资金窃取都将由AI自驱动完成，AI执行的所有过程与结果数据都能在平台中观测到。人类通过观测数据，辅助AI持续迭代完善安全知识与工具。

APG: 将红队专家经验编码为可执行的攻击思路图

APG(Attack Pattern Graph)是一种将攻击知识和专家经验编码为图结构的方法。通过节点(攻击步骤)和边(转移条件)来表示攻击流程,使AI能够理解和执行复杂的攻击策略。

01

图结构化

将攻击流程表示为有向图,节点代表攻击步骤,边代表转移条件。

02

可执行性

每个节点都包含可执行的攻击操作,AI可以直接调用和执行。

03

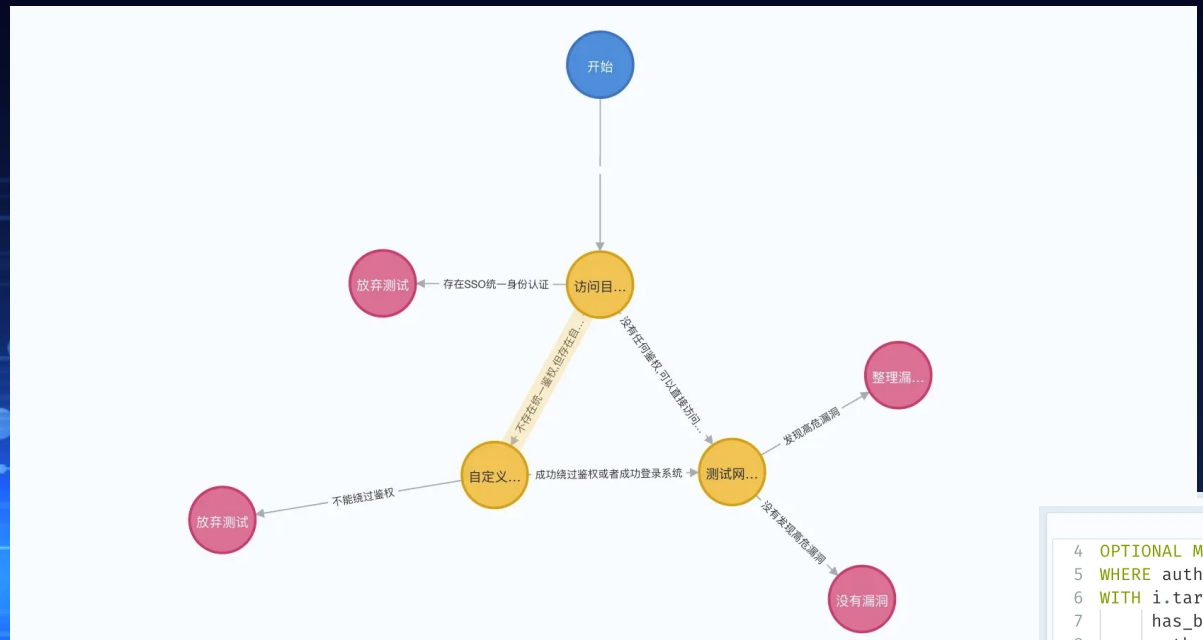
知识编码

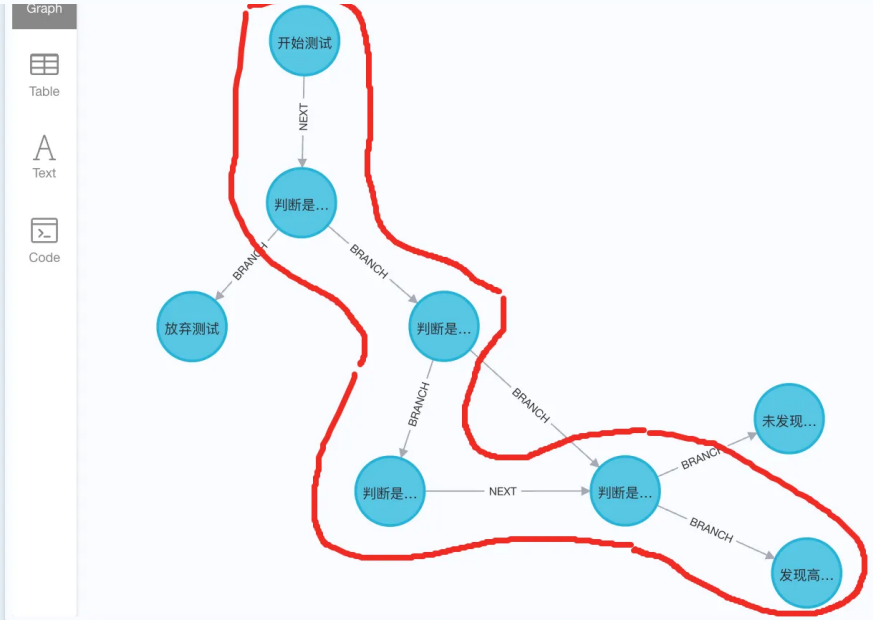
将红队专家的经验和知识编码为图的形式,便于复用和传播。

APG与传统方法论对比

维度	APG	ATT&CK	PTES
结构形式	图结构	矩阵/知识库	流程文档
可执行性	高(可直接执行)	低(参考性)	中(需人工适配)
AI友好度	高(结构化)	中(需转换)	低(非结构化)
自适应性	高(动态调整)	低(静态)	中(手工调整)
知识复用	高(图复用)	高(库复用)	中(流程复用)
应用场景	AI自动化攻防	威胁情报分析	渗透测试流程

APG的表示法与运行实例的数据查询






```cypher
OPTIONAL MATCH (i)-[:HAS_EXECUTION]→(auth:AIGExecution)
WHERE auth.apg_node_id = 'apg_node_004'
WITH i.target as 目标,
     has_buc,
     auth.custom_auth_detected as has_custom_auth,
     CASE
       WHEN has_buc THEN 'BUC认证'
       WHEN auth.custom_auth_detected THEN '自定义鉴权'
       ELSE '无鉴权'
     END as 认证类型
RETURN 认证类型, count(*) as 数量, collect(目标) as 系统列表
ORDER BY 数量 DESC;
```

Table | Text | Code

	认证类型	数量	系统列表
1	"自定义鉴权"	3	["example-office.company.com", "admin-portal.company.com", "dev-tools.company.com"]
2	"BUC认证"	2	["hr-system.company.com", "finance-dashboard.company.com"]
3	"无鉴权"	1	["legacy-app.company.com"]

Started streaming 3 records after 20 ms and completed after 22 ms.

APG Runtime: 执行引擎

APG作为静态图结构定义了专家经验的执行流程，但不能直接在AI模型上运行。**Runtime是APG的解释器(interpreter)和执行引擎(execution engine)**，在APG图定义与底层LLM之间扮演中间层角色。这里采用轻量级AI基础库（如pydantic-ai）而非高级Agent框架（如LangChain/LLamaIndex）。

图遍历(Graph Traversal)

维护当前执行状态,根据节点输出和边条件决定下一步转移

Prompt编译(Prompt Compiler)

将节点定义和上下文编译为可执行的AI提示

上下文管理(Context Engineering)

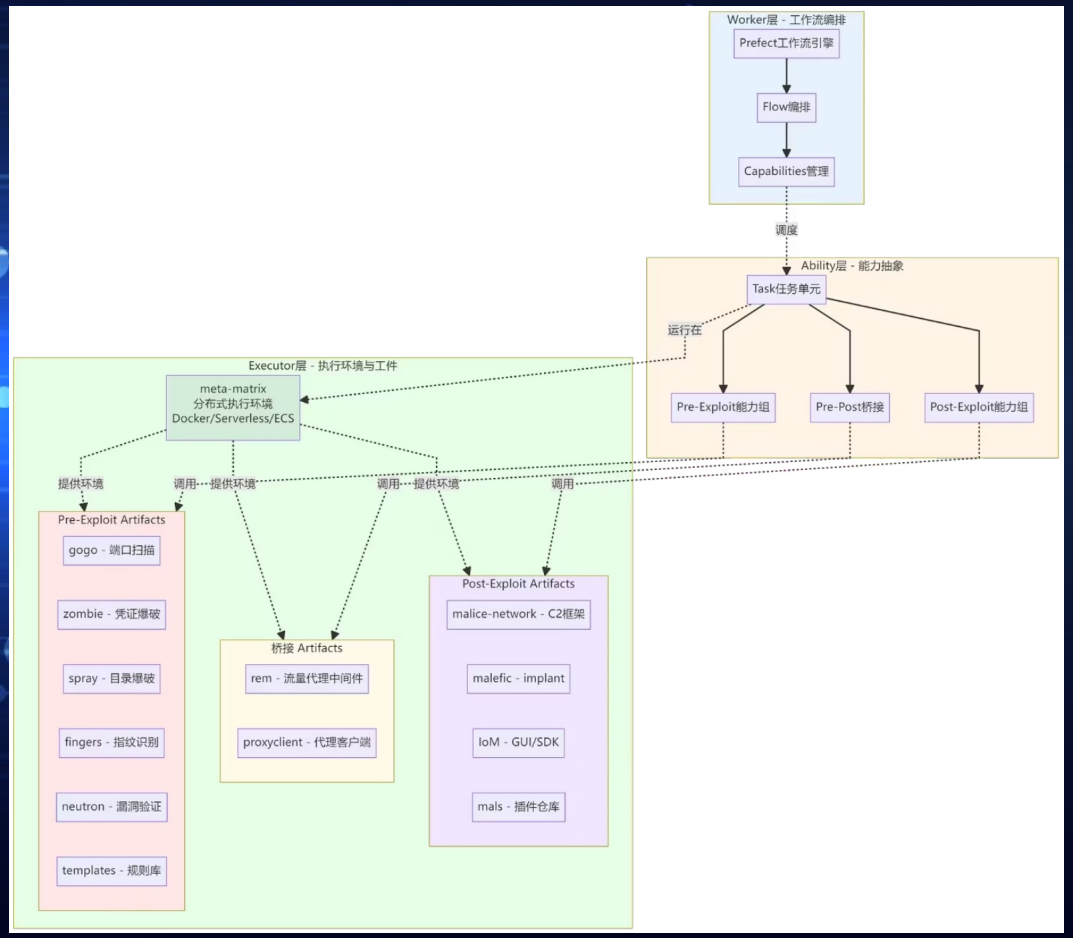
维护全局执行上下文,确保跨节点的信息连续性

工具调度(Tool Orchestration)

管理工具生态,按需注入工具能力

维度	HexStrike-AI	CAI	Strix	APG Runtime
架构模式	工具集成	Planner-Executor	多专业Agent	**图解释器**
执行模式	LLM自主调用工具	线性任务列表	Agent调度	**图遍历+动态编译**
上下文管理	全局工具可见	Planner/Executor隔离	Agent间信息传递	**结构化上下文**
专家知识组织	基于手册	Planner提示词	Agent提示词	**APG图结构**
适应性机制	规则驱动调理	重新规划(重新开始)	协调协议	异常驱动回溯+"图"展开

Offensive Infrastructure：AI的肌肉记忆



我们通过APG与APG Runtime尽可能保证AI能理解和执行我们的攻击意图。但是在攻防领域，安全工具能力的差异可能就是最后影响攻击效率甚至效果的最大因素。**我们不能让AI一次次的重新解决问题,因此我们要给AI提供足够强大的肌肉记忆。**

AI Native：AI时代的安全工具不会是Burpsuite、CobaltStrike这样的适合人类点击的图形化工具。AI更擅长编写代码解决问题，因此我们的所有基础设施都要进行**AI Native化改造。**

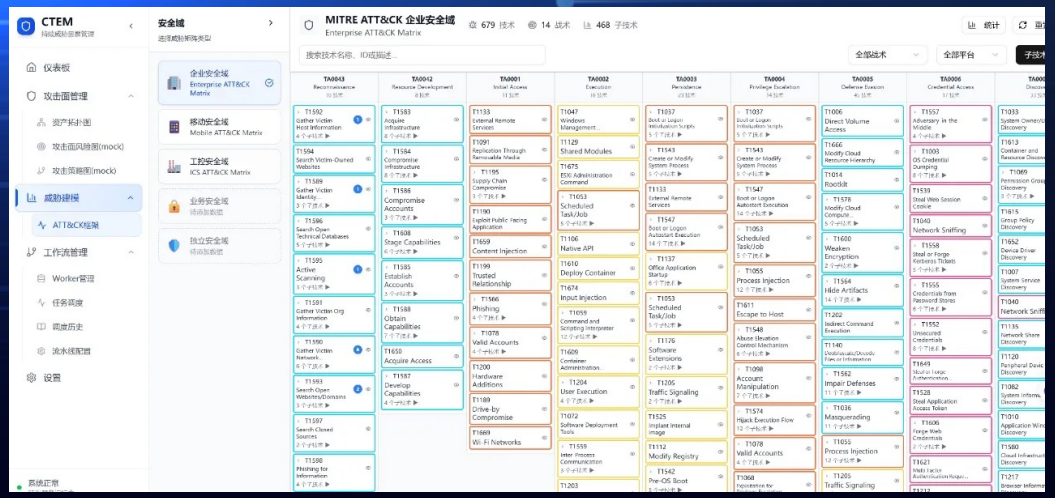
我们的实现方案是，**CodeMode 通过Python作为统一的操作入口**，让AI编写Python脚本去解决问题，而不是调用命令行、MCP、Tool等来回交互式的工具。
(CodeMode的理念，我们使用了让manus编写的大约100行代码在腾讯举办第一届AI智能渗透大赛就稳居前五)

AI时代的观测指挥平台

AI辅助人类，还是人类辅助AI?

在AI时代,不再需要一个复杂功能的操作平台,AI并不需要通过浏览器去进行一次次的点击,AI会直接调用Offensive Infrastructure完成所有的操作,那么我们的平台只需要用来监视AI正在做什么?怎么做的?结果是否符合预期。

我们的结论是,未来大概率是人类辅助AI完成绝大部分工作。因此我们的AI平台是一个AI可观测性平台,我们通过这个平台去了解AI正在做什么，而不是通过这个平台去操作AI。人类不断协助AI补充完善安全领域深度知识经验以及安全工具能力。



AI观测指挥平台，用于**监视AI正在做什么、怎么做的、结果是否符合预期，**而不是操作AI。未来大概率是**人类辅助AI**完成绝大部分工作。

① 监控 (Monitoring)

实时监视AI的行为和执行状态,了解AI正在执行什么操作。通过日志、指标和事件流,实现对AI执行过程的透明化。

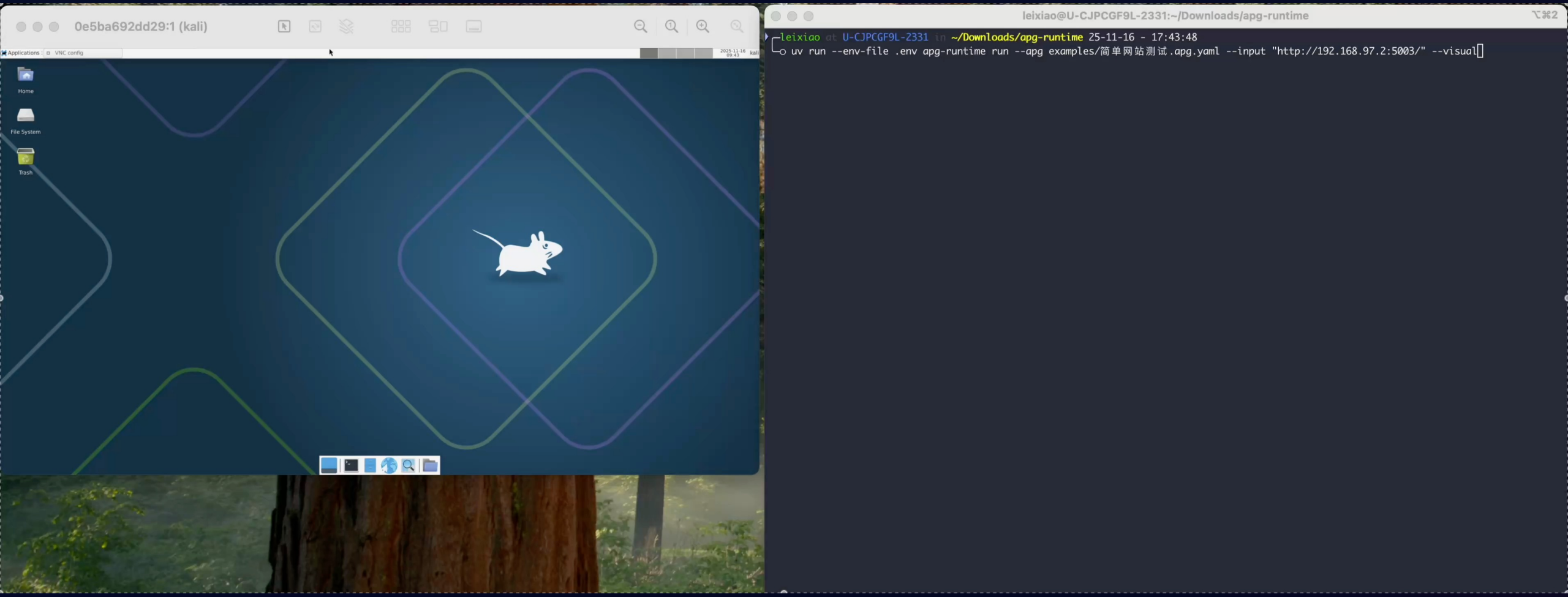
② 审计 (Auditing)

记录和审计AI的执行过程,追踪决策逻辑和行为路径。建立完整的审计日志,用于事后分析和合规性检查。

③ 反馈 (Feedback)

评估执行结果是否符合预期,提供改进建议和纠正机制。形成人-AI协作的反馈循环,持续优化AI的能力。

AI驱动自主实战网络攻击效果演示



AI实战攻击未来展望

历经多次大的迭代，已初步实现具备自主实战攻击能力。当前已应用在ASM、BAS以及办公网风险挖掘与利用等场景。

短期：**可复制的蓝军专家能力，替代50%+基础攻击工作**

沉淀各领域各等级安全专家经验

历史攻防手法、常见漏洞细节、各目标/阶段攻击思路

进攻型基础设施全面AI化

AI友好的攻击工具、真实的攻击能力、先进的攻击能力

可复制的中级蓝军安全专家能力

能够复现/检验/挖掘攻击链路，替代50%+工作

长期：**达到APT级别的攻击队能力，提升支付宝面向高等级攻击的常态防御能力。**

1、基于该架构快速提升各安全领域，各领域Agent形成协作。

我们在不同的安全子领域（代码审计、钓鱼社工、样本生成、业务理解、威胁感知等）也有很多AI驱动的实践，都可以升级为我们前面所讲的架构，每一个安全领域都能快速沉淀和提升，并可以在不同AI Agent间进行协作。

2、实现AI自迭代进化能力，拥有更快的学习成长。

再下一代AI将具有完全自主迭代能力,**接近人类顶级黑客/APT**,通过AI探索新的攻击路径,沉淀攻击经验。

1

将代码仓库暴露给AI,让AI能直接迭代我们的进攻性基础设施

2

对APG的自动化迭代、测试、验证循环,实现专家经验的自我更迭

3

与人类的无缝协作,通过自然语言去指挥队友一样自然

THANK YOU