

信息·趋势·感悟

THE POWER OF TIME, POWER OF C'S, WITNESS TO

美御 2024 漏洞研究院

暨第十八届信息安全高峰论坛
INFORMATION SECURITY CONFERENCE

关于AI驱动自动代码审计

的

实践与思考

漏洞研究院 冯骁韬



AI辅助漏洞检测方法发展演进



大模型时代（当下）

核心机制：业务逻辑理解
以及语义审计

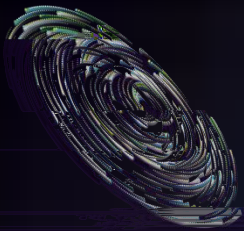
能力进化：通过语义推理的形式完成类似于人类代码审计的工作

核心机制：
表征提取

能力进化：在复杂结构中识别
非线性的漏洞模式

核心机制：
大规模特征匹配

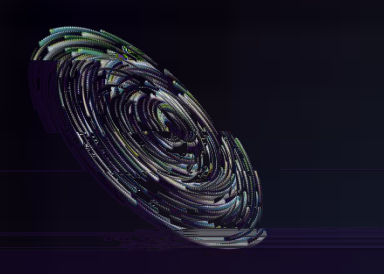
能力缺陷：无法理解代码的
上下文逻辑



目录

CONTENTS

- 1 Case Study - “逻辑自洽”的幻觉
- 2 AI代码审计落地的难题
- 3 方法论转型与Harness Engineering
- 4 总结



信息 + 趋势 + 感悟

THE POWER OF TIME, POWER OF C'S, WITNESS TO C'S

美盛 2024 年 12 月 24 日 研 究 报 告

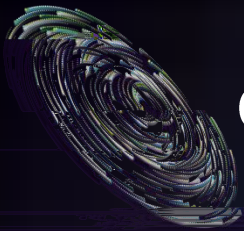
暨第十八届信息安全高峰论坛

INFORMATION SECURITY CONFERENCE

PART

01

Case Study “逻辑自治”的幻觉



Case Study - 一次由AI引导的漏洞审计

信息 · 趋势 · 感悟

THE POWER OF TIME, POWER OF GOD, WITNESS!

美赛 2024 网络安全研究

暨第十八届信息安全高峰论坛
INFORMATION SECURITY RESEARCH AND APPLICATION CONFERENCE

Claude Code + Opus-4-6

```

安全漏洞分析报告
● 存在高危路径遍历漏洞 (Path Traversal / Arbitrary File Write)

---
漏洞位置
文件:
行号: 第 158-162 行

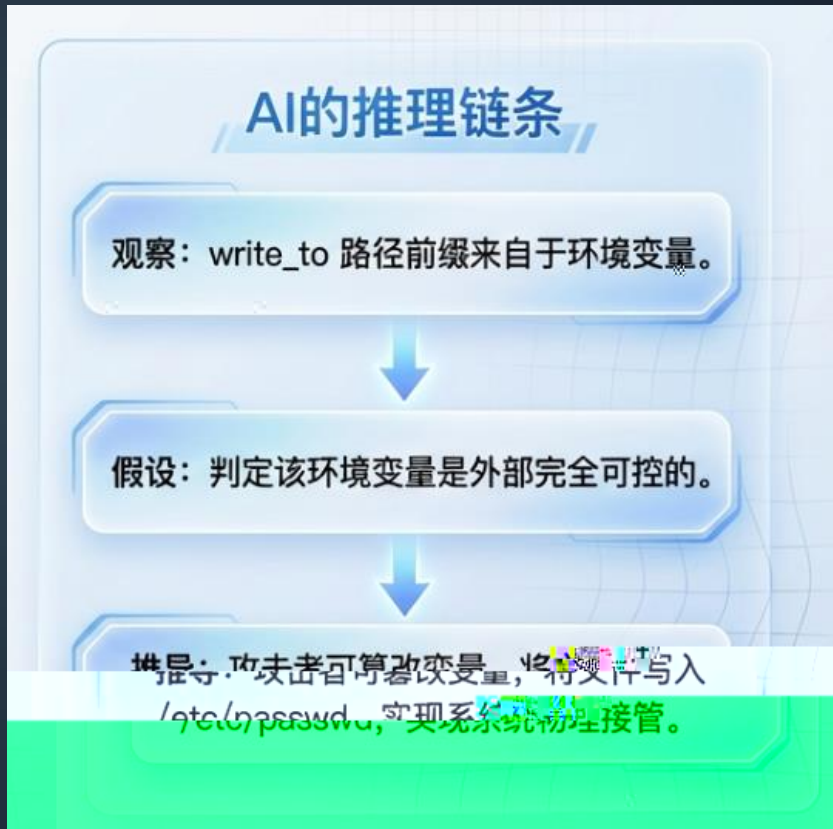
158 |         if write_to is True:
159 |             write_to = self.model_config.get("yaml_file")
160 |             if write_to:
161 |                 with open(write_to, "w") as f:
162 |                     f.write(yaml.dump(data))

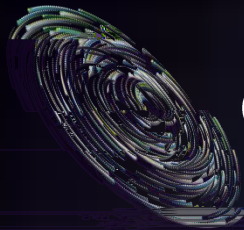
---
漏洞描述
1. 漏洞原理
- 第 161 行直接使用 write_to 参数作为文件路径，没有任何路径验证或清理
- write_to 参数可以来自:
  - 方法调用时传入的外部参数 (create_yaml_template(write_to=...))
  - 配置文件中的 model_config["yaml_file"] 值

2. 攻击场景
场景一: 直接调用攻击
template = YamlTemplate(model_obj)
# 攻击者控制 write_to 参数
template.create_yaml_template(write_to="/etc/passwd") # 覆盖系统文件
# 路径遍历
template.create_yaml_template(write_to="../../../sensitive/files.txt")

场景二: 配置注入攻击
如果攻击者能控制 model_config (例如通过配置文件注入)，可以设置:
model_config = {
    "yaml_file": "../../../etc/cron.d/malicious" # 写入定时任务
}

```





Case Study - “误报”

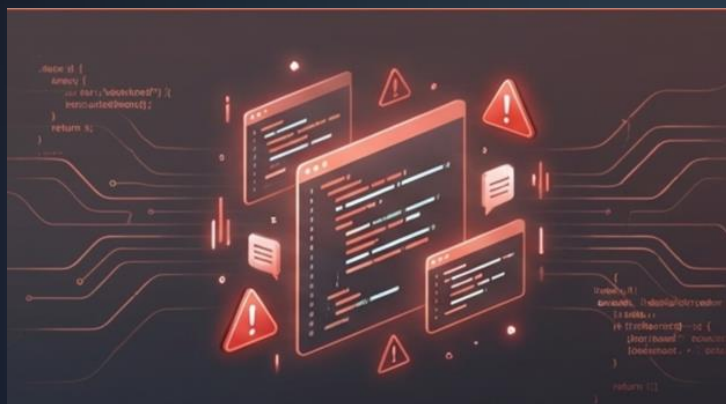
信息·趋势·感悟

THE POWER OF TIME, POWER OF C'S, WITNESS TO

美微 2024 中国 2024 高峰论坛

暨第十八届信息安全高峰论坛
INFORMATION SECURITY CONFERENCE

AI的“假设”

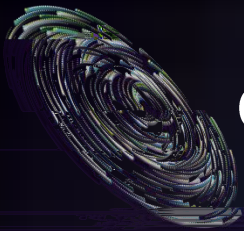


视角：纯代码级视野（语义推理）
判断逻辑：只要是外部输入，一律视为高危
结果：为了无“漏报”，产生“误报”

真实的部署环境



视角：架构与环境视野
真实环境：系统运行在受控的隔离环境
事实：系统运行在受控的隔离环境
在网络层面上无法进行干预



Case Study - “盲区”的本质 - AI模型被困于代码孤岛上

信息 + 趋势 + 感悟

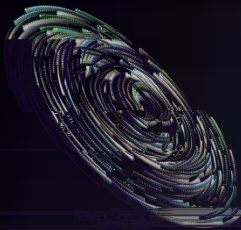
THE POWER OF TIME, POWER OF C'S, WITNESS!

美数 2024 中国 2024 高峰论坛

暨第十八届信息安全高峰论坛
INFORMATION SECURITY FORUM

THE ISOLATION EFFECT OF AI MODELS





信息 + 趋势 + 感悟

THE POWER OF TIME, POWER OF C'S, WITNESS TO C'S

美盛 2024 中国 AI 研究

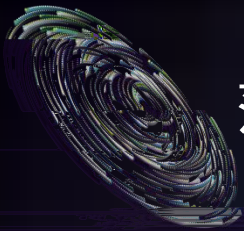
暨第十八届信息安全高峰论坛

INFORMATION SECURITY FORUM

PART

02

AI 代码审计落地的难题



落地的难题（二） - 通过提示词修复问题

漏洞的识别和修复Agent在判断漏洞时

需要注意部署环境

“在漏洞研判中,请考虑项目的部署情况”

Agent能够通过扫描项目的文档以及环境配置来思考外部攻击是否可以实现

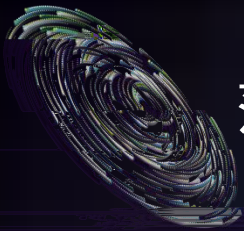
 ● 实际安全状态: 基本安全

关键事实:

- 方法调用场景
 - ✓ 只在 CLI 初始化命令 (python cli.py init) 中调用
 - ✓ 只在 脚本直接执行 时调用
 - ✗ 没有任何 API 端点暴露这些方法
- 参数来源
 - ✓ write to 参数完全硬编码为 True
 - ✓ 文件路径来自内部配置 (model_config["yaml_file"])
 - ✗ 无法从用户输入控制
- 部署场景

修正后

攻击类型	可行性	修复后
漏洞攻击	✗	● 不可行
命令注入	✗	● 不可行
服务端文件篡改	▲ 需要部署环境支持	● 不可行
本地文件读取	▲ 需要部署环境支持	● 不可行



落地的难题（三） - DeG生成需求

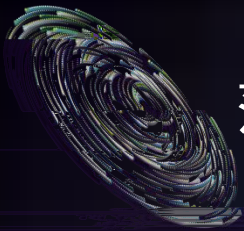
信息 + 趋势 + 感悟

THE POWER OF TIME, POWER OF GS WITH YOU!

美盛 2024 年 2 月 24 日 研研研

暨第十八届信息大会暨论坛
INFORMATION CONGRESS AND FORUM





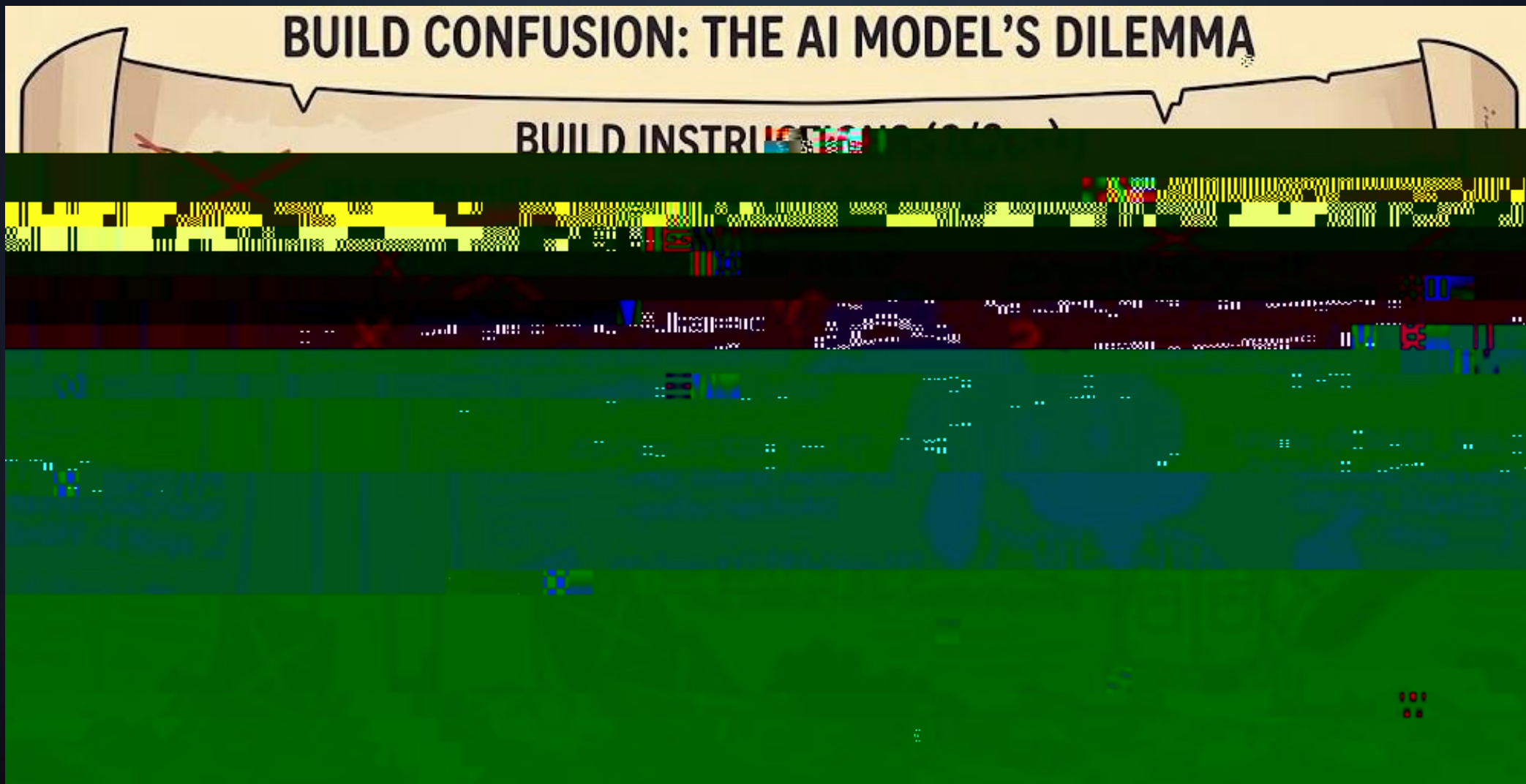
落地的难题（三） - BOC生成难题

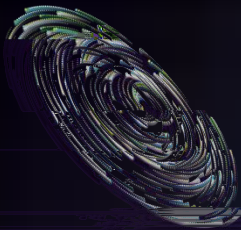
信息 + 趋势 + 感悟

THE POWER OF TIME, POWER OF C'S, WITH YOU!

美盛 2024 中国 AI 研究

暨第十八届信息大会暨论坛
INFORMATION TECHNOLOGY CONFERENCE





信息 + 趋势 + 感悟

THE POWER OF TIME, POWER OF C'S, WITH YOU!

美盛 2024 年 12 月 24 日 研 究 院

暨第十八届信息安全高峰论坛

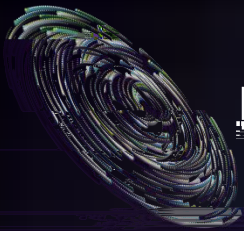
INFORMATION SECURITY CONFERENCE

PART

03

方法论转型 与

Harpass Engineering



Harness Engineering - Openai

信息 + 趋势 + 感悟

THE POWER OF TIME, POWER OF GOD, WITNESS!

美媒 MIT 美国 2024 年 1 月 研 究

暨第十八届信息安全高峰论坛

INFORMATION SECURITY CONFERENCE

Openai 团队将强大的模型比作拥有无限潜能，但是天性狂野难以控制的“烈马”

而Harness则是为这匹烈马量身定制的“马具”

Harness Engineering的本质就是旨在将模型的能力转化为受治理的可靠的行动

2026年2月11日 工程

工程技术：在智能体优先的世界中利用 Codex

作者: Dan Gohman | 技术 | openai.com | 2021年10月20日

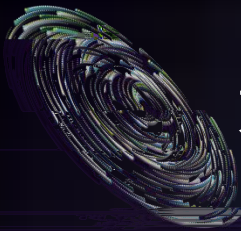
构建并交付一款软件产品的内部 beta 版，在过去五个月里，我们的团队一直在进行一项实验版，其中没有一行代码是人工编写的。

这产品有内部只供活跃用户和外部公开邀请者在过去五个月里，我们的团队一直在进行一项实验版，其中没有一行代码是人工编写的。

从应用逻辑、测试、CI 配置、文档可观察到内部工 过程。与众不同的，每一行代码 具 - 全都是由 Codex 编写的。据 成了这项工作。

人类掌舵。智能体执行。

我们有意选择这一限制，以便将 几周的时间来交付最终达到一百 主要工作，并显著地降低一下是 能够可靠地工作，会发 必要的内容，从而可 速度提升数千个量级。我们用了 行代码的 1/10 时间。我们需要了解，当软件 团队 在代码 准确性和稳定性。然而使 能够 做。



方法论的改变（一）

方法论的改变

从“模型驱动”转向“工程约束”：

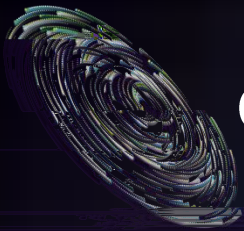
核心公式：智能体 = 模型 + 驾驭系统

底层模型是通用的，而决定审计质量的“护城河”是定制化的约束系统。我们不是去发明，应该通过设计精密的CAR（Control-Agency-Runtime）来构建。

破除“代码孤岛”——仓库即现实（Repo-as-truth）：

“只是模型在运行时无法访问的东西，对它而言就是不存在”

必须将所有的“隐性语义”显性化。除了代码，应将项目文档、设计文件、部署架构、环境变量约束以及历史补丁等全部写入知识库并进行索引



CAR

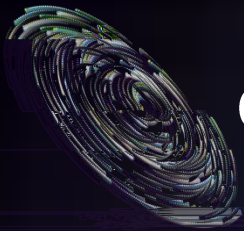
信息 + 趋势 + 感悟

THE POWER OF TIME, POWER OF CGS, WITNESS TO

美觀 20 萬圓 20 24 研研

暨第十八屆信息安全青年論壇

INFORMATION SECURITY YOUNG FORUM



CAR架构模型 - Agency 代理层

信息 + 趋势 + 感悟

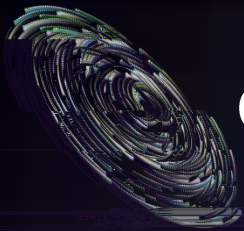
THE POWER OF TIME, POWER OF GOD, WISDOM OF...

美融 2016 中国 2016 高峰论坛

暨第十八届信息安全高峰论坛
INFORMATION SECURITY FORUM

Agency (代理层) : 引擎化分工与联动

停止让Agent进行海量且盲目的代码搜索 构建“新”



Runtime (运行时层) : 状态外部化与反馈自愈

状态外部化 (State Externalization)

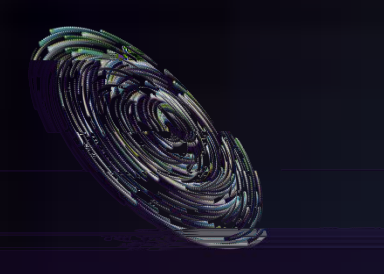
自愈闭环 (Feedback Loop)

问题。

• 承认幻觉：在工程上接受模型输出中的 DoC 失真

• 模型校准：在推理时动态调整 DoC 阈值

• 置信度管理：根据任务复杂度动态调整 DoC 阈值



信息 + 趋势 + 感悟

THE POWER OF TIME, POWER OF CGS, WITNESS TOY!

美盛文化2024年年报

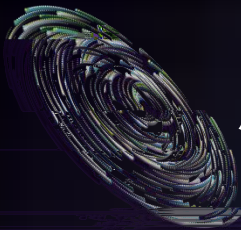
暨第十八届信息安全高峰论坛

INFORMATION SECURITY FORUM

PART

04

总结



总结

信息 + 趋势 + 感悟

THE POWER OF TIME, POWER OF CGS WITHIN YOU!

美御 2024 安全研究

暨第十八届信息安全高峰论坛
INFORMATION SECURITY CONFERENCE

核心观点：Agent = Model + Harness

视角转变：不再把Agent当作独立的“专家”，而是将其作为整个“安全审计系统”中的一个特定引擎

安全审计系统的CAR架构：

Control: 定义边界，引入文档，Patch 验证与修复

Agent: 赋予能力，引擎化分工，多工具 (SAST、Linter、沙盒) 联动

Runtime: 确保持续化运行，状态外部化 (知识库) 解决成本问题，自愈闭环 (日回传) 解决幻觉问题。

未来的核心竞争力不再是如何写好Prompt，而是构建严谨、可审计、自愈的Harness Engineering体系

信息·趋势·感悟

THE POWER OF TIME, POWER OF C'S, WITNESS!

美融 2010 中国 2024 高峰论坛

暨第十八届信息安全高峰论坛
INFORMATION SECURITY FORUM

THANKS!

