

PR #47595 完整报告

vllm-project/vllm

fix(security): guard lm-format-enforcer regex compile with timeout

合并时间: 2026-07-14 15:18

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/47595>

执行摘要

- 一句话: 修复 lm-format-enforcer 正则编译超时漏洞
- 推荐动作: 建议合并。这是一次轻量但必要的安全补丁, 填补了之前安全修复的遗漏, 代码简洁且模式成熟。

功能与动机

lm-format-enforcer 后端在 GHSA-rwxx-mrjm-wc2m 安全修复中被遗漏, 未对正则表达式编译施加超时保护, 存在 ReDoS 风险。PR body 明确提到 'The lm-format-enforcer backend was left unguarded after the fix for GHSA-rwxx-mrjm-wc2m'。

实现拆解

1. 导入超时工具函数: 在 vllm/v1/structured_output/backend_lm_format_enforcer.py 顶部添加 `from vllm.v1.structured_output.utils import compile_regex_with_timeout`。
2. 替换 RegexParser 调用: 在 `compile_grammar` 方法的 REGEX 分支中, 将直接构造 `lmformatenforcer.RegexParser(grammar_spec)` 替换为 `compile_regex_with_timeout(lmformatenforcer.RegexParser, grammar_spec)`, 使正则编译受超时保护。
3. 提前验证正则: 在 `validate_structured_output_request_lm_format_enforcer` 函数中, 当 `so_params.regex` 存在时, 先调用 `compile_regex_with_timeout` 进行编译, 若抛出异常则转换为 `ValueError`, 实现早期失败, 避免请求进入后续处理后才暴露问题。

关键文件:

- vllm/v1/structured_output/backend_lm_format_enforcer.py (模块 前端; 类别 source; 类型 dependency-wiring): 唯一修改文件, 添加超时保护和提前验证

关键符号: `compile_grammar`, `validate_structured_output_request_lm_format_enforcer`

关键源码片段

vllm/v1/structured_output/backend_lm_format_enforcer.py

唯一修改文件, 添加超时保护和提前验证

```
# 新增导入: 从工具模块引入带超时的正则编译函数
from vllm.v1.structured_output.utils import compile_regex_with_timeout
```

```

class LMFormatEnforcerBackend(StructuredOutputBackend):
    def compile_grammar(self, request_type, grammar_spec):
        # ...
        elif request_type == StructuredOutputOptions.REGEX:
            # 原来 : character_level_parser = lmformatenforcer.RegexParser(grammar_spec)
            # 现在 : 通过 compile_regex_with_timeout 增加超时保护, 防止 ReDoS
            character_level_parser = compile_regex_with_timeout(
                lmformatenforcer.RegexParser,
                grammar_spec,
            )
        # ...

def validate_structured_output_request_lm_format_enforcer(params: SamplingParams):
    # ...
    if so_params.regex:
        # 新增: 在请求验证阶段就尝试编译正则, 提前捕获无效或恶意模式
        try:
            compile_regex_with_timeout(
                lmformatenforcer.RegexParser,
                so_params.regex,
            )
        except Exception as err:
            raise ValueError(
                f"Failed to compile regex for lm-format-enforcer: {err}"
            ) from err
    return

```

评论区精华

无实质性 review 讨论: claude bot 因 PR 来自 fork 自动跳过审查, khluu 直接批准。

- 暂无高价值评论线程

风险与影响

- 风险: 低风险。变更范围极小 (仅 1 个文件, +14/-1), 且调用的是已存在的工具函数 `compile_regex_with_timeout`, 该函数已在 `xgrammar` 和 `outlines` 后端中经过验证。唯一潜在风险是 `compile_regex_with_timeout` 在极端情况下可能对合法但复杂的正则造成误杀, 但超时阈值应已考虑典型使用场景。
- 影响: 影响范围仅针对使用 `lm-format-enforcer` 后端的用户, 且仅在处理正则表达式时生效。对于不使用 `structured output` 或使用其他后端 (`xgrammar`、`outlines`) 的用户无影响。安全收益明确, 可防范 ReDoS 攻击。
- 风险标记: 微小变更, 低影响

关联脉络

- 暂无明显关联 PR