PR #47260 完整报告

vllm-project/vllm

fix(security): add resource bounds validation to derender endpoints

合并时间：2026-07-07 14:58

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47260

执行摘要

- 一句话：为 derender 端点添加资源边界验证，防止资源耗尽。
- 推荐动作：值得精读，展示了如何在端点添加安全校验的简洁模式。设计决策上，将验证前置到早期处理，复用现有配置项，是良好的实践。

功能与动机

derender 端点未对用户提供的 GenerateResponse 对象进行资源限制，攻击者可构造超长 token_ids 或大量 choices 导致 CPU/ 内存耗尽（CWE-400/CWE-770）。正常生成路径已有类似限制，derender 端点需要补齐。

实现拆解

1. 在 vllm/entrypoints/scale_out/token_in_token_out/protocol.py 的 GenerateResponseChoice 中添加 field_validator validate_token_ids，在 Pydantic 解析层直接拒绝负数 token_ids。
2. 在 vllm/entrypoints/scale_out/derender/serving.py 的 ServingDerender 类中新增 _validate_derender_bounds() 方法，依次检查 generate_responses 数量、choices 数量、token_ids 长度、logprobs.content 长度、top_logprobs 数量、prompt_logprobs 长度，超过对应上限时返回 ErrorResponse。
3. 在 derender_chat_response 和 derender_completion_response 方法开头注入 bounds_error 检查，在调用 derender 之前快速返回错误。
4. 在 tests/entrypoints/scale_out/derender/test_derender.py 中添加集成测试，覆盖正向（正常负载通过）和所有越界场景（token_ids 超长、choices 过多、generate_responses 过多、负数 token_ids、logprobs 超长、top_logprobs 过多）。

关键文件：

- vllm/entrypoints/scale_out/derender/serving.py（模块 反渲染服务；类别 source；类型 core-logic；符号 _validate_derender_bounds）：核心变更文件，新增 _validate_derender_bounds 方法，并在两个端点入口调用。
- vllm/entrypoints/scale_out/token_in_token_out/protocol.py（模块 协议定义；类别 source；类型 core-logic；符号 validate_token_ids）：在 GenerateResponseChoice 中添加 field_validator 拒绝负数 token_ids。

- tests/entrypoints/scale_out/derender/test_derender.py（模块 测试；类别 test；类型 test-coverage；符号 test_derender_chat_bounded_payload_succeeds, test_derender_chat_oversized_token_ids_rejected, test_derender_chat_too_many_choices_rejected, test_derender_completion_too_many_generate_responses_rejected）：添加完整的边界测试套件，覆盖正反场景。

关键符号：_validate_derender_bounds, validate_token_ids, derender_chat_response, derender_completion_response

关键源码片段

vllm/entrypoints/scale_out/derender/serving.py

核心变更文件，新增 _validate_derender_bounds 方法，并在两个端点入口调用。

```
def _validate_derender_bounds(
    self,
    generate_responses: list[GenerateResponse],
) -> ErrorResponse | None:
    """Reject derender payloads that exceed resource bounds.

    Runs before any tokenizer.decode() or parser invocation to prevent
    CPU/memory exhaustion from oversized caller-supplied token structures.
    """
    max_n = envs.VLLM_MAX_N_SEQUENCES
    max_model_len = self.model_config.max_model_len

    # 检查 generate_responses 数量 （Completions 端点多条）
    if len(generate_responses) > max_n:
        return self.create_error_response(
            f"generate_responses count ({len(generate_responses)}) "
            f"exceeds server maximum ({max_n}). "
            f"Set VLLM_MAX_N_SEQUENCES to increase this limit."
        )

    for gen in generate_responses:
        # 每个 GenerateResponse 的 choices 数量
        if len(gen.choices) > max_n:
            return self.create_error_response(
                f"choices count ({len(gen.choices)}) in response "
                f"'{gen.request_id}' exceeds server maximum ({max_n})."
            )

        for choice in gen.choices:
            # token_ids 长度不超过 max_model_len
            if choice.token_ids and len(choice.token_ids) > max_model_len:
                return self.create_error_response(
                    f"token_ids length ({len(choice.token_ids)}) in "
                    f"choice {choice.index} exceeds "
```

```
                f"max_model_len ({max_model_len})."
            )
        # logprobs.content 长度不超过 max_model_len
        if choice.logprobs and choice.logprobs.content:
            if len(choice.logprobs.content) > max_model_len:
                return self.create_error_response(
                    f"logprobs.content length "
                    f"({len(choice.logprobs.content)}) in "
                    f"choice {choice.index} exceeds "
                    f"max_model_len ({max_model_len})."
                )
            # 每个 top_logprobs 数量不超过 20 （OpenAI 规范限制）
            for entry in choice.logprobs.content:
                if entry.top_logprobs and len(entry.top_logprobs) > 20:
                    return self.create_error_response(
                        f"top_logprobs count "
                        f"({len(entry.top_logprobs)}) in "
                        f"choice {choice.index} exceeds maximum (20)."
                    )

    # prompt_logprobs 长度不超过 max_model_len
    if gen.prompt_logprobs and len(gen.prompt_logprobs) > max_model_len:
        return self.create_error_response(
            f"prompt_logprobs length ({len(gen.prompt_logprobs)}) "
            f"in response '{gen.request_id}' exceeds "
            f"max_model_len ({max_model_len})."
        )

    return None
```

评论区精华

DarkLight1337 询问 top_logprobs 上限 20 的来历，作者说明来自 OpenAI 规范，但当前未在代码或文档中显式注明；DarkLight1337 同意另开 PR 处理，不阻塞此 PR 合并。

- top_logprobs 上限 20 的文档化 (design): 同意另开 PR 补充文档，当前不阻塞。

风险与影响

- 风险：
  - 兼容性风险：新的验证可能拒绝某些合理的大请求，但限制值与正常生成路径一致，用户可通过调整 VLLM_MAX_N_SEQUENCES 适应。
  - 文档风险：magic number 20 未在代码注释或文档中说明来源，可能引起困惑。
  - 性能风险：验证开销极小，无显著影响。
  - 安全风险：有效缓解资源耗尽攻击，安全性提升。
  - 影响：用户使用 derender 端点将受到新限制，越界请求返回 400 错误。系统安全性提升，资源耗尽风险降低。团队后续需补充关于限制值的文档说明。

- 风险标记：安全边界检查，配置文档缺失

关联脉络

- 暂无明显关联 PR