PR #45252 完整报告

vllm-project/vllm

[Security] Fix DoS via prompt_embeds on M-RoPE models

合并时间：2026-06-13 18:17

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/45252

执行摘要

- 一句话：修复 M-RoPE 模型 prompt_embeds 拒绝服务漏洞
- 推荐动作：建议精读，该 PR 展示了典型的安全修复模式：将硬断言转为优雅处理。代码逻辑清晰，测试覆盖完整，值得作为类似问题的参考。关注点：_init_mrope_positions 中的三路分支设计和测试中通过 Mock 和 object.__new__ 绕过初始化的技巧。

功能与动机

当向 M-RoPE 模型发送仅包含 prompt_embeds 的请求（如 /v1/completions with prompt=None + prompt_embeds）时，原代码中的 assert req_state.prompt_token_ids is not None 会触发 AssertionError，导致 EngineCore 崩溃，形成拒绝服务（DoS）漏洞。该漏洞由安全公告 GHSA-33cg-gxv8-3p8g 报告。PR body 明确说明需要“Replace the fatal assertion in _init_mrope_positions that crashes the EngineCore when prompt_token_ids is None”。

实现拆解

1. 移除硬断言，改为条件分支：在 vllm/v1/worker/gpu_model_runner.py 的 _init_mrope_positions 方法中，移除了 assert req_state.prompt_token_ids is not None 断言。新增三个分支处理：如果 prompt_token_ids 非空则使用它；否则如果 prompt_embeds 非空则从 prompt_embeds.shape[0] 获取序列长度并生成虚拟 token ID；否则抛出 ValueError。这样做是因为 M-RoPE 的 get_mrope_input_positions 在无多模态特征时仅需输入长度，而 prompt_embeds 已经被过滤掉，因此虚拟 ID 是安全的。
2. 更新调用参数：将 get_mrope_input_positions 的参数从 req_state.prompt_token_ids 改为 input_tokens（新变量），以支持两种输入来源。
3. 新增测试文件：tests/v1/worker/test_mrope_prompt_embeds.py 是一个全新的测试文件，包含一个伪造的 M-RoPE 模型 FakeMRoPEModel 和一个辅助函数 _make_runner_and_req，用于创建最小化的 GPUModelRunner 实例和请求状态。测试类 TestMRopePromptEmbeds 包含三个测试用例：test_prompt_embeds_only_does_not_crash 验证仅 prompt_embeds 不会崩溃；test_prompt_token_ids_still_works 验证正常路径仍可用；test_neither_token_ids_nor_embeds_raises 验证两者均为 None 时抛出正确的 ValueError。

关键文件：

- vllm/v1/worker/gpu_model_runner.py（模块 模型运行器；类别 source；类型 core-logic；符号 _init_mrope_positions）：核心修改文件：修复 _init_mrope_positions 方法中的 DoS 漏洞，将断言改为优雅的三路分支处理。
- tests/v1/worker/test_mrope_prompt_embeds.py（模块 M-RoPE 位置；类别 test；类型 test-coverage；符号 FakeMRoPEModel, get_mrope_input_positions, _make_runner_and_req, TestMRopePromptEmbeds）：新增测试文件，全面覆盖修复逻辑的三个分支：仅 prompt_embeds、仅 prompt_token_ids、两者均无。

关键符号：_init_mrope_positions

关键源码片段

vllm/v1/worker/gpu_model_runner.py

核心修改文件：修复 _init_mrope_positions 方法中的 DoS 漏洞，将断言改为优雅的三路分支处理。

```
def _init_mrope_positions(self, req_state: CachedRequestState):
    model = self.get_model()
    assert supports_mrope(model), "M-RoPE support is not implemented."
    mrope_model = cast(SupportsMRoPE, model)

    # Filter out prompt_embeds: they are a passthrough modality with no grid_thw.
    # M-RoPE models assume per-feature grid info, so we exclude them.
    mrope_features = [
        f for f in req_state.mm_features if f.modality != "prompt_embeds"
    ]

    # Deciding input_tokens:
    # - If prompt_token_ids exist, use them directly.
    # - If only prompt_embeds exist, generate dummy token IDs based on sequence length.
    # This is safe because get_mrope_input_positions only needs len(input_tokens)
    # when mm_features is empty (which is the case after filtering above).
    # - If neither exists, raise a clear ValueError.
    if req_state.prompt_token_ids is not None:
        input_tokens = req_state.prompt_token_ids
    elif req_state.prompt_embeds is not None:
        # prompt_embeds is a 2D tensor: (seq_len, embedding_dim)
        seq_len = req_state.prompt_embeds.shape[0]
        input_tokens = list(range(seq_len))
    else:
        raise ValueError(
            "M-RoPE requires either prompt_token_ids or prompt_embeds."
        )

    req_state.mrope_positions, req_state.mrope_position_delta = (
        mrope_model.get_mrope_input_positions(
            input_tokens,
            mrope_features,
```

```
        )
    )
```

tests/v1/worker/test_mrope_prompt_embeds.py

新增测试文件，全面覆盖修复逻辑的三个分支：仅 prompt_embeds、仅 prompt_token_ids、两者均无。

```
# SPDX-License-Identifier: Apache-2.0
# SPDX-FileCopyrightText: Copyright contributors to the vLLM project
"""Test that M-RoPE position initialization handles prompt_embeds-only inputs.

Regression test for GHSA-33cg-gxv8-3p8g: sending /v1/completions with
prompt_embeds and no prompt_token_ids on M-RoPE models crashed the
EngineCore via an assertion failure.
"""

from unittest.mock import Mock

import pytest
import torch

from vllm.model_executor.models.interfaces import SupportsMRoPE
from vllm.v1.worker.gpu_input_batch import CachedRequestState
from vllm.v1.worker.gpu_model_runner import GPUModelRunner


class FakeMRoPEModel(SupportsMRoPE):
    """Minimal model that passes supports_mrope() check."""

    def get_mrope_input_positions(self, input_tokens, mm_features):
        seq_len = len(input_tokens)
        positions = torch.arange(seq_len).unsqueeze(0).expand(3, -1)
        return positions.clone(), 0


def _make_runner_and_req(prompt_token_ids, prompt_embeds):
    """Create a minimal GPUModelRunner instance and request state."""
    model = FakeMRoPEModel()
    instance = object.__new__(GPUModelRunner)
    instance.get_model = lambda: model

    req_state = Mock(spec=CachedRequestState)
    req_state.prompt_token_ids = prompt_token_ids
    req_state.prompt_embeds = prompt_embeds
    req_state.mm_features = []
    req_state.mrope_positions = None
    req_state.mrope_position_delta = None
    return instance, req_state
```

```
class TestMRopePromptEmbeds:
    """Verify _init_mrope_positions handles prompt_embeds-only inputs."""

    def test_prompt_embeds_only_does_not_crash(self):
        """Prompt-embeds-only request must not raise AssertionError."""
        instance, req_state = _make_runner_and_req(
            prompt_token_ids=None,
            prompt_embeds=torch.randn(15, 896),
        )
        instance._init_mrope_positions(req_state)
        assert req_state.mrope_positions is not None
        # Expect (3, 15) shape: 3 axes for M-RoPE, seq_len=15
        assert req_state.mrope_positions.shape == (3, 15)

    def test_prompt_token_ids_still_works(self):
        """Normal path with prompt_token_ids continues working."""
        instance, req_state = _make_runner_and_req(
            prompt_token_ids=[1, 2, 3, 4, 5],
            prompt_embeds=None,
        )
        instance._init_mrope_positions(req_state)
        assert req_state.mrope_positions is not None
        assert req_state.mrope_positions.shape == (3, 5)

    def test_neither_token_ids_nor_embeds_raises(self):
        """When both are None, a ValueError should be raised."""
        instance, req_state = _make_runner_and_req(
            prompt_token_ids=None,
            prompt_embeds=None,
        )
        with pytest.raises(ValueError, match="prompt_token_ids or prompt_embeds"):
            instance._init_mrope_positions(req_state)
```

评论区精华

评论数量很少，只有一位审核者询问是否需要额外内容。没有深入的代码审查讨论。两位审核者（qthequartermasterman 和 DarkLight1337）均直接批准了 PR。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。变更范围小，仅修改了单个函数的控制流，且通过新增的三个测试用例覆盖了所有分支。但需注意：当 prompt_embeds 非空且 mm_features 不为空时，虚拟 token ID 的长度可能不匹配多模态特征所需的信息；不过当前代码中 prompt_embeds 已被过滤掉，所以对 M-RoPE 调用不会传入 mm_features，风险很小。回退路径明确：如果两者均为 None，则抛出 ValueError 而不是静默失败。

- 影响：影响范围有限：仅影响使用 M-RoPE 的模型（如部分多模态模型），且仅涉及处理仅包含 prompt_embeds 请求的路径。对正常使用 prompt_token_ids 的请求无影响。修复了安全漏洞（DoS），提高了服务的稳定性。
- 风险标记：安全漏洞修复，控制流变更，新增测试覆盖

关联脉络

- PR #45467 [Model Runner V2] Fix openai.InternalServerError: Error code: 500 - 'list index out of range': 同属 Model Runner 的 bugfix，与 Model Runner 内部状态处理相关。
- PR #45347 [BugFix] Avoid prematurely freeing cached mm encoder outputs: 同为 v1 调度器 /Model Runner 相关的 bugfix，涉及请求状态管理。