

PR #52952 完整报告

vllm-project/vllm

[Bugfix][Security] Guard `_load_ov2_processor` with `resolve_trust_remote_code`

合并时间: 2026-08-20 17:09

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/52952>

执行摘要

- 一句话: 修复 OV2 处理器 `trust_remote_code` 被静默忽略的安全漏洞
- 推荐动作: 值得精读, 因为它演示了如何正确使用 `resolve_trust_remote_code` 来加固远程代码加载, 对维护涉及 `trust_remote_code` 的模型加载逻辑有参考价值。建议后续补充针对 `trust_remote_code=False` 的自动化测试。

功能与动机

PR body 指出, `_load_ov2_processor` 将 `trust_remote_code` 传给了 `get_class_from_dynamic_module`, 但该函数没有此参数, 值被 `**kwargs` 静默吸收并丢弃。这导致即使 `trust_remote_code=False`, 攻击者提供的处理模块仍会执行, 构成安全风险。

实现拆解

1. 修改导入语句: 在 `vllm/model_executor/models/llava_onevision2.py` 中, 从 `transformers.dynamic_module_utils` 导入新增的 `resolve_trust_remote_code` 函数, 与已有的 `get_class_from_dynamic_module` 并列导入。
2. 添加守卫调用: 在 `_load_ov2_processor` 函数体内, 调用 `convert_model_repo_to_path` 和设置 `revision` 之后、调用 `get_class_from_dynamic_module` 之前, 插入 `resolve_trust_remote_code` 调用, 传入 `trust_remote_code`、模型名, 并显式声明 `has_local_code=False`, `has_remote_code=True`, 强制校验信任策略。
3. 移除多余的参数传递: 删除 `get_class_from_dynamic_module` 调用中的 `trust_remote_code=trust_remote_code` 参数, 因为这些参数本会被忽略, 移除后代码意图更清晰, 也避免误导。
4. 未添加测试: PR 未包含测试文件变更, 仅依赖手动验证清单。

关键文件:

- `vllm/model_executor/models/llava_onevision2.py` (模块 模型加载; 类别 `source`; 类型 `data-contract`): 核心修复文件, 在处理器加载逻辑中添加安全守卫

关键符号: `_load_ov2_processor`

关键源码片段

`vllm/model_executor/models/llava_onevision2.py`

核心修复文件，在处理器加载逻辑中添加安全守卫

```
# vllm/model_executor/models/llava_onevision2.py
# 在 _load_ov2_processor 函数中，新增安全校验，确保 trust_remote_code=False
# 时不会加载远程代码

# 导入新增的 resolve_trust_remote_code
from transformers.dynamic_module_utils import (
    get_class_from_dynamic_module,
    resolve_trust_remote_code,
)

@lru_cache
def _load_ov2_processor(
    model: str,
    revision: str | None,
    trust_remote_code: bool,
    **kwargs: Any,
):
    # ... 其他初始化代码
    path = convert_model_repo_to_path(model)
    revision = revision or "main"

    # 关键修复：在加载远程处理器类之前，显式校验 trust_remote_code 策略
    # 若 trust_remote_code=False 且仓库含远程代码，这里会抛出 ValueError，阻止后续导入
    resolve_trust_remote_code(
        trust_remote_code,
        model,
        has_local_code=False, # 明确无本地代码
        has_remote_code=True, # 明确存在远程代码
    )

    # 远程处理器类加载，不再传入无效的 trust_remote_code 参数（原参数被静默忽略）
    processor_cls = get_class_from_dynamic_module(
        "processing_llava_onevision2.LlavaOnevision2Processor",
        path,
        revision=revision,
    )
    video_processor_cls = get_class_from_dynamic_module(
        "video_processing_llava_onevision2.LlavaOnevision2VideoProcessor",
        path,
        revision=revision,
    )
    # ... 后续处理器组装逻辑
```

评论区精华

由于该 PR 来自 fork，Claude bot 自动评论禁用，无实际 code review 讨论。维护者 DarkLight1337 直接批准（APPROVED），无评论。Issue 评论区仅有 CI 触发记录。

- 暂无高价值评论线程

风险与影响

- 风险：功能回归风险：对于 `trust_remote_code=True` 的合法用户，`resolve_trust_remote_code` 调用不会改变行为，但需验证其是否与原有逻辑完全兼容，尤其是对 `has_local_code` / `has_remote_code` 的判断是否正确。若某些模型仓库结构特殊（如既有本地代码又有远程代码），可能导致错误拦截。兼容性风险：`resolve_trust_remote_code` 是 transformers 的相对较新函数，旧版本可能不存在，可能导致依赖兼容问题。但 vllm 通常会固定 transformers 版本，风险较低。测试缺失：未添加自动化测试，无法防止未来回归或验证边界情况（如本地代码存在时行为）。
- 影响：用户影响：修复了潜在的安全漏洞，对于使用 `trust_remote_code=False` 且处理器从远程加载的用户，现在会正确抛出 `ValueError` 并阻止执行，避免攻击者注入的恶意代码运行。系统影响：影响范围限于 `llava_onevision2` 模型的处理器加载路径，不会影响其他模型或核心推理路径。团队影响：贡献者需确保与 transformers API 变更同步，并考虑后续补充测试。
- 风险标记：安全修复，缺少测试覆盖

关联脉络

- 暂无明显关联 PR