PR #21435 完整报告

sgl-project/sglang

[Security] 1/N: Bind ZMQ sockets to localhost to prevent unauthenticated remote access

合并时间：2026-03-27 14:33

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/21435

执行摘要

本 PR 通过将 ZMQ sockets 默认绑定到 localhost（127.0.0.1），有效阻止了未认证远程访问，缓解了三个 CVSS 9.8 的 CVE 漏洞（CVE-2026-3059、CVE-2026-3060、CVE-2026-3989），这些漏洞源于不安全 pickle.loads() 反序列化，可导致远程代码执行。变更覆盖多模态生成 ZMQ broker 和编码器分散模块，通过操作系统内核在数据到达前拒绝连接，提升了系统安全性，同时保持跨机器访问的向后兼容性。

功能与动机

PR 的动机直接源于三个关键安全漏洞：CVE-2026-3059 影响多模态生成 ZMQ broker，CVE-2026-3060 影响编码器并行分散，CVE-2026-3989 影响脚本组件。这些漏洞允许攻击者通过未认证的 ZMQ sockets 进行 pickle.loads() 反序列化，导致远程代码执行（RCE）。引用 PR body 中的表述："The OS kernel rejects remote TCP connections before any data reaches pickle.loads()"，目标是防止远程攻击，同时通过绑定到 localhost 来限制网络可达性，作为第一道防线。

实现拆解

实现分为三个核心模块的改动：

文件路径	变更内容	目的
python/sglang/multimodal_gen/runtime/scheduler_client.py	将 broker_endpoint 从 tcp://*:{broker_port} 改为 tcp://127.0.0.1:{broker_port}	解决 CVE-2026-3059，限制 ZMQ broker 仅本地访问
python/sglang/srt/disaggregation/encode_receiver.py	在两个调用 get_zmq_socket_on_host 的地方显式传递 host 参数	适配 CVE-2026-3060，确保跨机器访问时正确配置
python/sglang/srt/utils/network.py	修改 get_zmq_socket_on_host 函数，默认 host 从 None 改为 127.0.0.1，并更新文档说明安全考虑	核心安全变更，影响所有 ZMQ sockets 的默认绑定行为

关键代码逻辑：在 network.py 中，函数 get_zmq_socket_on_host 现在默认绑定到 localhost：

```
if host is None:
    host = "127.0.0.1"
```

这确保了在没有显式 host 参数时，sockets 仅监听本地接口。

评论区精华

review 讨论中最有价值的交锋集中在设计假设上。kpham-sgl 在 scheduler_client.py 行 20 评论："1. It seems like broker_port is hardcoded to port + 1. Do we know why? 2. Is this ZMQ broker intended to be consumed only by clients within the same host?"，这触及了安全修复的底层设计权衡——默认绑定到 localhost 是否会影响分布式用例。尽管未看到直接回复，但 Fridge003 的批准表明团队接受了这一变更，可能基于现有上下文或安全优先原则。讨论强调了在安全修复中验证设计假设的重要性。

风险与影响

风险具体包括：一是默认值变更可能导致历史分布式部署失败，如果代码中遗漏 host 参数传递；例如，其他未修改的调用 get_zmq_socket_on_host 的地方可能意外受限。二是跨机器访问场景需额外配置，增加了运维复杂度。影响分析：安全性大幅提升，直接缓解 RCE 风险；性能无显著变化；兼容性方面，通过显式参数保持向后兼容，但需团队更新相关文档和测试以确保所有用例适配。

关联脉络

从历史 PR 分析看，PR #20904（修复 CVE-2026-3989）与本 PR 高度相关，两者都属于同一安全修复链，旨在解决由不安全反序列化引起的 CVE。这表明仓库近期正集中处理安全漏洞，趋势是加强默认安全配置和替换危险函数。本 PR 作为第一环，通过网络隔离降低攻击面，为后续更深入的反序列化修复（如 PR #20904 中的 SafeUnpickler）奠定基础，共同构建更健壮的安全防线。