

基于eBPF的容器异常检测 框架与方法项目进展汇报

指导教师：任怡 赵欣

小组成员：毕喜舒 马永媛 刘周康

参赛队编号：T202590002995558

时间：2025. 6. 26

01

项目参赛队员分工

02

项目开发时间进度安
排及任务推进情况

01

项目参赛队员分工

负责模块

负责eBPF探针开发与优化，包括进程、系统调用、文件、TCP等探针的编写与调试，确保数据采集的准确性和高效性。

任务推进

已完成多个核心eBPF探针的开发，如进程追踪、文件操作监控等，能够实时捕获容器内关键事件数据。

预期成果

进一步优化探针性能，降低资源消耗，提升数据采集的完整性和实时性，为后续异常检测提供高质量数据支持。

01

负责模块

主导容器元信息管理模块设计与实现，构建容器与进程的映射关系，确保容器元数据的准确获取与实时更新。

02

任务推进

完成了容器信息数据结构设计及用户态数据处理逻辑，能够准确追踪容器内进程并实时更新容器元信息。

03

预期成果

持续完善容器元信息管理模块，优化数据存储与查询效率，为系统提供稳定可靠的容器元数据支持，助力异常检测与分析。

负责模块

负责安全规则设计与实现，构建基于规则的异常检测引擎，同时协助AI检测模块的集成与优化。

任务推进

已初步搭建安全规则框架，实现部分基础安全规则的定义与检测逻辑，为系统安全告警提供基础支持。

预期成果

深入调研容器安全风险场景，完善安全规则库，提升异常检测的准确性和覆盖面，同时优化AI检测模块与规则引擎的协同工作。

02

项目开发时间进度安排
及任务推进情况

阶段一：赛题理解与技术储备（1-2周）（已完成）

eBPF技术学习

完成《BPF Performance Tools》前3章实操，深入理解eBPF技术原理及应用，掌握CO-RE特性。

容器运行时监控研究

搭建Kubernetes测试环境，使用docker stats和cadvisor分析容器资源指标，熟悉容器监控核心概念。

异常检测算法对比

对比孤立森林、LSTM、AutoEncoder等算法，确定项目采用孤立森林算法作为异常检测核心算法。

阶段二：数据采集框架开发（3-4周）（已完成）



eBPF探针设计与开发

使用libbpf开发框架，完成多个核心eBPF探针的编写，捕获系统调用、容器指标等关键数据。



数据管道构建

设计环形缓冲区存储事件数据，实现用户态数据处理程序，构建高效的数据采集管道。



性能优化

添加过滤机制减少无效事件，测试不同map类型性能，优化数据采集框架性能。

阶段三：异常检测算法开发（3-4周）（已完成）

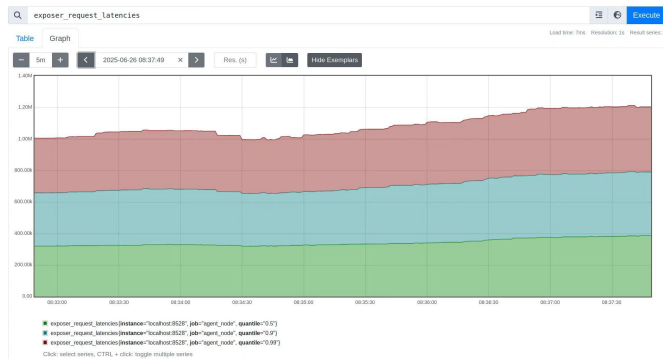
基线算法实现

实现基于统计的阈值检测算法，开发规则引擎，完成异常进程检测、端口扫描检测等基础功能。

机器学习模块开发

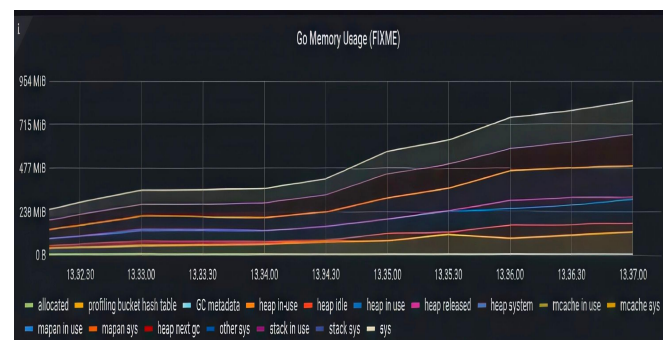
完成特征工程，从原始事件提取时序特征；使用sklearn实现孤立森林模型训练，初步完成模型部署。

阶段四：可视化系统集成（1-2周）（已完成）



数据抓取监控

使用Prometheus抓取数据，对数据实施实时监控与故障诊断，将分散的系统指标转化为可追溯的时间序列数据。



数据看板开发

使用Grafana搭建监控仪表盘，设计集群概览、容器实例详情、异常事件时间线等展示维度。

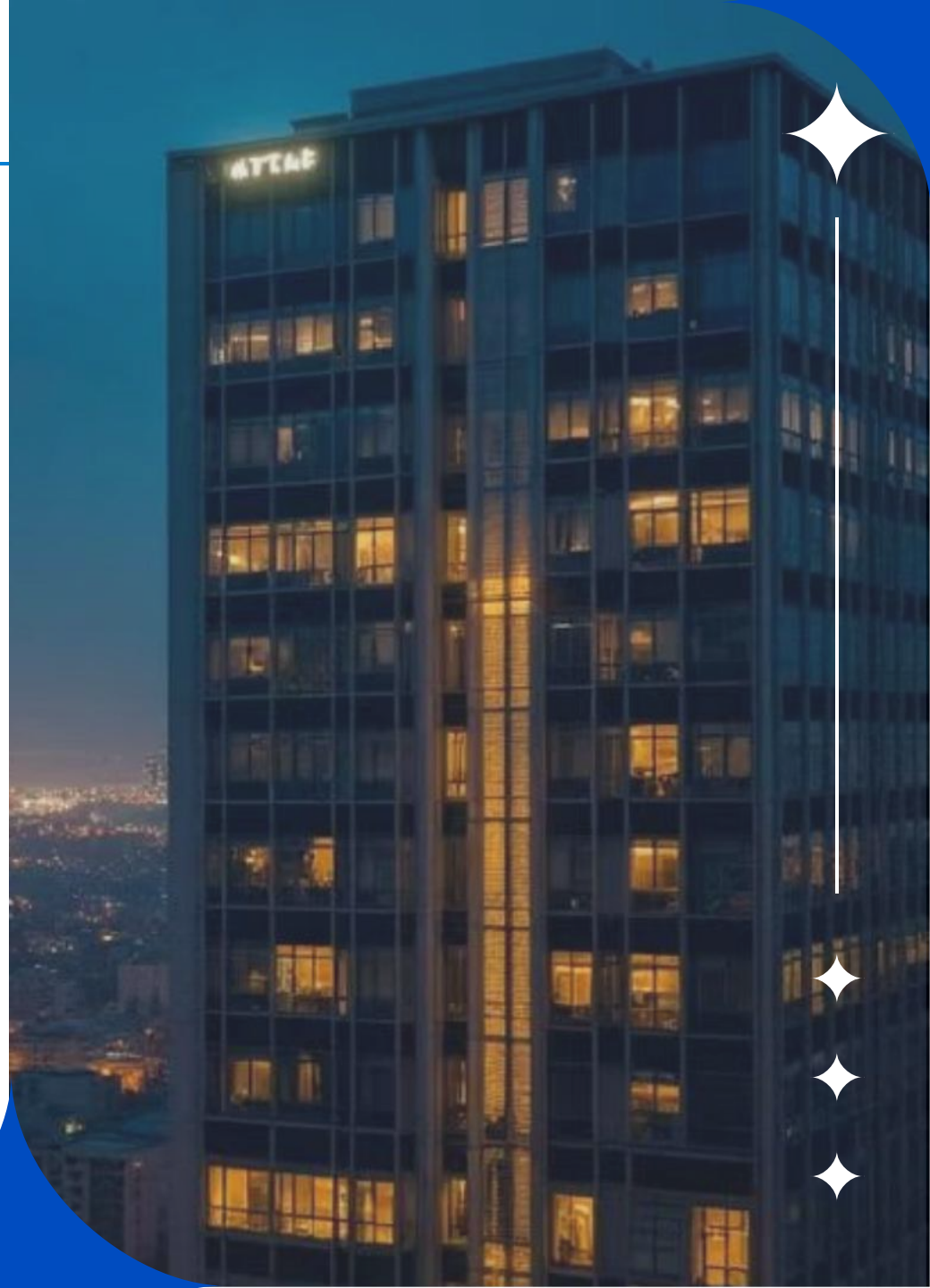
阶段五：扩展功能开发（2-3周）

性能比较框架设计

设计基准测试场景，开发性能对比工具，评估不同采集工具及检测算法的性能表现。

自愈能力实现

集成Kubernetes API，实现异常容器自动隔离；开发补救策略引擎，提供分级响应机制。



阶段六：系统联调与优化（2周）

全链路压力测试

使用k6工具模拟高并发容器场景，进行72小时稳定性测试，确保系统在高负载下的稳定运行。

安全加固

增加eBPF程序验证机制，实现权限分级控制，提升系统整体安全性。

阶段七：文档与交付（1周）（阶段性文档交付已完成）

成果视频拍摄

介绍仓库运行环境，演示项目编译运行过程，结合Prometheus和grafana进行可视化展示。

文档编写

编写详细的API文档，生成技术白皮书，涵盖系统架构图、核心算法说明及性能对比测试报告等内容。

感谢各位专家批评指正！

汇报小组：三个臭皮匠

时间：2025. 6. 26

