吴逸飞 Wu Yifei
yinwhe@zju.edu.cn | www.yinwhe.me | 17799857027

完整信息

教育背景

浙江大学（硕士） **2023.09-2026.03**

网络空间安全（CS Rank Asia No.1, World No.11）导师：申文博

- 顶刊 TDSC（CCF-A）学生一作一篇
- 顶会 ICSE（CCF-A）二作一篇，顶刊 TSE（CCF-A）二作一篇
- IJIS（ESORICS Best Paper Invited Extension）一作一篇
- 2025 小米奖学金，2024 / 2025 浙江大学优秀研究生 & 学年学业优秀奖助金一等奖

浙江大学（本科） **2019.8 –2023.06**

信息安全 GPA：4.71/5.0（Rank 2）

- 省政府奖学金 2020
- 全国大学生系统能力竞赛内核赛优胜奖 2021
- 华为智能基座奖学金 2022
- 优秀本科毕业论文 & 优秀毕业生 2023

技术关键字

- 系统软件、操作系统、eBPF、容器、Rust；

主要经历

模型可观测&性能分析 **2025.05-2025.09**

阿里巴巴 基础设施实习 A+评级

- 模型性能 Roofline & Simulation：对各类 Qwen 模型计算过程建模，分析计算图及各个算子的维度等信息，利用 Roofline / Simulation 建模对算子的性能进行分析并整体合并，指导集团内存量软硬件匹配、硬件演进以及未来优化方向。
- 模型性能 CPU & GPU 联合分析：开发低开销无侵入的 Python / C++ Profiler、基于 CUPTI 的 GPU Profiler，对其两者数据获取完整的 CPU & GPU Timeline 性能视图，辅助模型性能问题定位、性能分析以及优化指导。
- 训练 Hang / Slow 节点检测：基于 eBPF Uprobe 监测模型训练 Steps，快速定位训练 Hang 节点，同时触发 CPU & GPU 联合分析能力，保存问题现场，利于后续问题分析与定位。

Rust 编译器不稳定特性的全面研究 **2022.03-2023.11**

软工领域顶会 ICSE 二作一篇，顶刊 TSE 二作一篇 主要 Rust 编写，LoC 1.6w

- RUF 提取器：利用 Rustc 生成的语法树，并重定向数据流以避免过早的编译失败和不必要的 I/O 操作，高效地提取 RUF 使用情况。揭露了不稳定特性发展的规律和存在的异常问题，精确提取了生态内不稳定特性的使用情况。
- 生态系统解析器：采用虚拟解析环境和容器构建生态系统解析器，在 AlderLake 机器上一天内处理 2.48 亿个传递依赖项并拥有 99% 的准确率和召回率。基于生态解析，我们发现至多 44% 生态被不稳定特性影响，至多 12% 无法编译。
- 依赖树修复：利用自研修复算法，结合版本语义验证、RUF 兼容性分析和包预分析，可高效修复 91% 的编译失败问题。

抽象资源攻击的理解与防御：eBPF 在内核监控里的应用 **2023.11-2024.05**

顶刊 TDSC 学生一作一篇

- 新攻击面：揭示了抽象资源攻击新攻击面，可以通过消耗系统的结构、变量等实现 DoS。抽象资源攻击具有很强的实用性，能影响 Linux、FreeBSD 和 Fuchsia 等系统，且已经在 AWS、MS Azure 等平台攻击实验成功。
- Kernel IR 分析：基于 LLVM 设计并实现了一个静态分析工具，用于识别 Linux 内核中的易受攻击的抽象资源，检测出 Linux 内核中有 501 个可动态且反复触发的抽象资源。
- eBPF 防御框架：基于 eBPF 开发了监控框架，通过将 eBPF 探针附加到工具分析出的资源分配函数，采用基于树的资源核算方法，以及 eBPF Filter 和 Cache 技术，Flask 能够高效地限制抽象资源（低于 1% 的开销）。

硬件支持的隔离机制：以 eBPF 为例 **2024.03-2024.12**

专利审核中

- 隔离框架：基于 ARM 硬件特性的进程内隔离框架，将安全风险组件（例如 JIT 编译代码、第三方插件等）解耦到硬件隔离的单元中，同时保持本地执行效率。该框架允许跨多个域主动遏制内存损坏、数据泄漏和权限提升风险。
- eBPF 案例：实现 eBPF 子系统安全增强的应用案例，通过强制的内存隔离限制了 eBPF 对关键内存组件（映射表、栈和上下文）的访问，降低潜在恶意 eBPF 带来的风险，开销控制在 4% 以内。

eBPF 劫持攻击与低开销防御 **2025.02-2025.08**

IJIS（ESORICS Best Paper Invited Extension）一作一篇

- Interpret / JIT 有效攻击：提出 Tailcall Trampoline 利用技术，以一个完全无特权的普通用户身份，在 eBPF Interpreter 和 JIT 上分别实现任意代码执行和任意内核内存访问，进一步实现提权等攻击。
- 低开销防御：在 eBPF 执行路径上插入基于 IDR 的指针合法性验证，以 1% 的开销成功防御攻击。

技能/语言

4 年 Rust 语言经验，熟悉 C/C++，有基本的 Go、Python、SQL、Web3 项目经验，CET6 554。

More

华为 OpenEuler 研发实习 **2021.12-2022.06**

Dagrs 开发：一个基于 Rust 的高性能异步任务框架，采用流式编程（FBP）简化多任务编排和通信。

- 通过抽象任务，应用程序能够高效扩展并降低异步编程的复杂度，框架以模块化“黑盒”组件构建网络，灵活配置数据流，同时严格控制数据生命周期和资源使用，支持有向无环图（DAG）依赖编排，内置有限容量连接器实现智能拥塞控制。
- 主要技术栈包括：Rust（性能与可靠性）、Tokio（非阻塞 I/O）和 async_trait（简化异步接口实现）。
- 目前在 GitHub 上获得超过 400 颗星，作为最初开发者，其核心代码仍在使用。

Rust 系统与跨语言安全 **2023.02-2024.12**

优秀本科毕业论文 深入分析 Rust 与 C/C++ 等语言之间的交互问题，探讨从低层 ABI 不兼容到高层语义匹配的多维度挑战，实现 ABI 交互匹配分析原型工具，提前检测出交互错误；研究生阶段，在此基础上系统分析了 Rust for Linux 的架构设计和 API 安全性，研究 Rust 与 C 在内核绑定与互操作性问题，并尝试将现有检测工具扩展至 R4L 生态，以提高系统整体安全性。

基于 Rust 编写的容器引擎 **2024.09-2025.02**

Rtain 是一个完全用 Rust 语言开发的容器管理引擎，包括超过 2500 行的 Rust 代码，采用客户端-守护进程架构，提供了一套完整的容器生命周期管理解决方案，不依赖 RunC 等现有框架。

- 容器操作：支持大多数基本的容器功能，包括创建、运行、执行、镜像生成和网络配置。
- 独立实现：从底层重写，避免了对 RunC 等框架的依赖，提供了一个独特且优化的容器运行时环境。
- 本地存储：集成了具备预写日志（WAL）和快照功能的本地文件存储系统，确保高效的状态管理和恢复。
- 高并发：采用 Tokio 异步框架处理高并发工作负载，极大提高容器管理吞吐量。

RISC-V64 操作系统 hJH OS **2021.03-2021.06**

队长 hJH OS 使用 C 语言编写，运行于 kendryte-K210 双核处理器，支持 FAT32 虚拟文件系统以及 27 条系统调用，**获得全国大学生计算机系统能力培养大赛操作系统内核赛道优胜奖 。**

- 系统调用支持：实现了约 27 个系统调用，涵盖进程管理、内存管理、文件处理以及一些其他实用功能。
- 文件系统支持：支持 FAT32 虚拟文件系统，内置原生 SD 卡集成和文件块缓存优化功能，以提升性能。