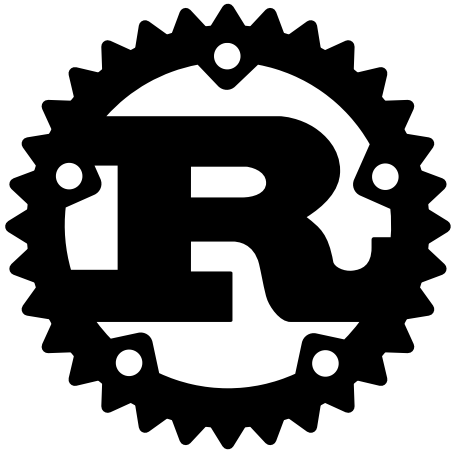
Riddler - 网络分析与HTTP工具

高效的网络监控、HTTP分析与性能诊断工具



By Farewe1ll

项目概述

Riddler是一个多功能网络分析工具，使用Rust语言开发，主要功能包括：

- 🔍 **网络监控** - 实时捕获和分析网络流量
- 🌐 **HTTP 请求分析** - 详细解析 HTTP/HTTPS 请求和响应
- 🍪 **Cookie 管理** - 智能处理和持久化存储 Cookie
- 📊 **性能分析** - 诊断长响应时间问题
- 🔁 **请求重放** - 自动重放监控到的HTTP请求
- 🔌 **代理服务器** - 提供HTTP/HTTPS代理功能

核心技术栈

- **Tokio** - 高性能异步运行时
- **pcap/pnet** - 网络数据包捕获与解析
- **reqwest** - HTTP客户端请求发送
- **clap** - 命令行界面构建
- **serde** - 数据序列化与反序列化
- **DashMap** - 并发安全的哈希表

功能详解：网络监控

- **多平台适配** - 支持 Linux、macOS 和 Windows
- **BPF 过滤器** - 精确筛选需要监控的流量
- **实时 HTTP 解析** - 从 TCP 数据包中还原 HTTP 请求
- **权限管理** - 自动检测和提示权限需求

```
if let Some(network_packet) = Self::parse_packet(packet.data) {
    let is_potential_http = network_packet.dst_port == 80 ||
                network_packet.dst_port == 443 ||
                HttpParser::contains_http_method(&network_packet.payload);
    if is_potential_http {
    }
}
```

功能详解：性能分析

- **多维度指标采集** - DNS 解析、TCP 连接、TLS 握手等
- **瓶颈智能诊断** - 自动识别性能瓶颈所在
- **专业响应时间分类** - 从 Excellent 到 Critical 的 5 级分类
- **针对性优化建议** - 根据问题特征提供解决方案

```
fn determine_severity(&self, total_time_ms: u64) -> PerformanceSeverity {
    match total_time_ms {
        0..=100 => PerformanceSeverity::Excellent,
        101..=500 => PerformanceSeverity::Good,
        501..=1000 => PerformanceSeverity::Average,
        1001..=3000 => PerformanceSeverity::Poor,
        _ => PerformanceSeverity::Critical,
    }
}
```

功能详解：请求重放与日志管理

- **智能请求重放** - 可配置重放次数、间隔和过滤条件
- **结构化日志存储** - JSON 格式保存所有请求细节
- **高级搜索与过滤** - 支持多条件查询和统计分析
- **持久化Cookie管理** - 自动处理 Cookie 生命周期

实现挑战与解决方案

挑战一：跨平台网络接口兼容性

- **挑战**: 不同操作系统有不同的网络接口命名和权限管理
- **解决方案**:
  - 自动检测操作系统类型并提供默认配置
  - 实现接口自动发现和列举

```
let default_interface = match std::env::consts::OS {
    "macos" => "en0",
    "linux" => "eth0",
    "windows" => "<请用--interface参数指定网络接口>",
    _ => "en0",
}.to_string();
```

实现挑战与解决方案

挑战二：HTTP解析的健壮性

- **挑战**: 从TCP流中准确提取HTTP请求
- **解决方案**:
  - 实现多阶段解析和验证
  - 使用启发式方法识别HTTP方法
  - 处理不完整和分段的HTTP数据

挑战三：性能分析精准度

- **挑战**: 精确测量各阶段响应时间
- **解决方案**:
  - 使用多次迭代测试增加可靠性
  - 智能区分网络和服务器延迟

其他的一些困难。。。

- `SIGINT` `SIGTERM` `SIGKILL` 复合退出程序
- `Request` 库自动检测代理
- `http` 请求解析不正确

项目演示

基本功能展示

1. 网络监控与 HTTP 请求捕获
2. 请求发送与 Cookie 管理
3. 性能分析与诊断
4. 请求回放与数据收集

技术总结

Riddler 项目通过 Rust 语言的高性能和安全特性，结合现代网络分析技术，提供了一套完整的 HTTP/HTTPS 请求监控、分析和诊断工具。

谢谢!