用树莓派打造家庭广告拦截系统

叶家炜

May 05, 2025

现代网络广告不仅影响浏览体验，更通过跟踪脚本与恶意广告威胁用户隐私。传统浏览器插件方案存在覆盖范围有限（无法保护智能电视等设备）、配置繁琐等痛点。基于树莓派构建的 DNS 层广告拦截系统，通过单点部署即可实现全网络设备覆盖，结合开源工具 Pi-hole 的过滤能力，以接近零的边际成本构建家庭级隐私防护屏障。

1 项目概述

系统核心原理是通过 DNS 协议拦截广告域名解析请求。当设备发起网络访问时，树莓派上的 Pi-hole 会优先检查域名是否存在于广告黑名单中。若命中规则则返回空响应阻断连接，否则将请求转发至上游 DNS 服务器完成正常解析。相较于传统方案，该架构具备网络层拦截优势，可覆盖路由器、游戏主机等无法安装插件的设备。硬件推荐使用 Raspberry Pi 4B（2GB 内存版本），其 1.5GHz 四核处理器与千兆网口可轻松应对家庭网络吞吐需求。软件栈以 Raspberry Pi OS Lite 为基础，通过 Pi-hole 提供广告过滤功能，可选搭配 Unbound 实现本地递归 DNS 解析以进一步提升隐私性。

2 准备工作

树莓派基础配置需优先完成网络连接与系统优化。使用 Raspberry Pi Imager 刷写系统时，建议启用 SSH 并预配置 Wi-Fi 连接信息。关键步骤是设置静态 IP 地址，避免因 DHCP 分配变动导致服务中断。通过修改 `/etc/dhcpcd.conf` 文件实现：

```
interface eth0
static ip_address=192.168.1.100/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

此配置将树莓派的以太网接口固定为 `192.168.1.100`，子网掩码 `/24` 对应 255.255.255.0，网关与 DNS 指向路由器地址。系统优化阶段需执行 `sudo apt update && sudo apt upgrade -y` 更新软件源，并安装 `curl` 用于获取 Pi-hole 安装脚本。

3 安装与配置 Pi-hole

通过官方提供的一键安装脚本部署 Pi-hole：

```
curl -sSL https://install.pi-hole.net | bash
```

该命令通过 curl 下载安装脚本并交由 bash 解释执行。安装过程中需注意两个关键选项：

- 上游 **DNS** 选择：推荐使用 Cloudflare（1.1.1.1）或 Quad9（9.9.9.9）等支持 DNSSEC 的服务商
- 管理界面密码：安装完成后会生成随机密码，可通过 pihole -a -p 命令修改

广告列表订阅建议导入 Steven Black 维护的统一 hosts 列表，该列表聚合了多个优质规则源。在 Web 管理界面（可通过 http://树莓派 IP/admin 访问）的 **Group Management > Adlists** 中添加以下 URL：

```
https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts
```

4 网络设备配置

在路由器管理界面将默认 DNS 服务器设置为树莓派的静态 IP。以 TP-Link Archer 系列为例，进入 网络 **> DHCP** 服务器 页面，修改 主 **DNS** 服务器 与 备用 **DNS** 服务器 字段。对于不支持全局修改的路由器，需在终端设备手动配置 DNS：

1. **Windows**：打开「网络和 Internet 设置」>「更改适配器选项」> 右键属性 > IPv4 属性
2. **Android**：进入 Wi-Fi 设置 > 长按当前网络 > 修改网络 > 勾选「高级选项」> IP 设置改为静态

5 高级功能扩展

集成 Unbound 可将树莓派升级为本地递归 DNS 服务器，避免向上游服务商发送查询请求。安装后需修改 Pi-hole 的上游 DNS 配置指向本地 5335 端口：

```
sudo apt install unbound
echo 'server:␣interface:␣127.0.0.1␣port:␣5335' | sudo tee /etc/unbound/unbound.conf.d
    ↪ /pi-hole.conf
```

Unbound 通过迭代查询从根域名服务器自主完成解析，其响应时间 $T_{response}$ 可表示为：

$$T_{response} = T_{root} + T_{TLD} + T_{authoritative}$$

其中 T_{root} 为根服务器查询延迟，T_{TLD} 为顶级域名服务器延迟，$T_{authoritative}$ 为权威服务器延迟。实际测试显示首次查询耗时约 200-300ms，后续因缓存机制可降至 10ms 以内。

6 常见问题与优化

广告过滤失效时，首先检查客户端 DNS 缓存。Windows 系统执行 ipconfig /flushdns 强制刷新，Linux 使用 systemd-resolve --flush-caches。若遇误拦截，在 Pi-hole 管理界面的 **Whitelist** 添加域名即可。

性能优化建议将日志存储于内存磁盘。创建 tmpfs 挂载点并修改 Pi-hole 配置：

```
echo 'tmpfs␣/var/log/␣tmpfs␣defaults,noatime,nosuid,size=50m␣0␣0' | sudo tee -a /etc/
    ↪ fstab
sudo systemctl restart pihole-FTL
```

此配置将日志写入内存，减少 SD 卡写入损耗。公式推导显示，假设日均日志量 D_{log} 为 100MB，使用 `tmpfs` 后 SD 卡写入量降低比例 R_{reduce} 为：

$$R_{reduce} = 1 - \frac{D_{log}}{D_{total}} = 1 - \frac{100}{D_{total}}$$

当 D_{total} 包含系统写入时，实际延长 SD 卡寿命约 3-5 倍。

实测数据显示，典型家庭网络环境下 Pi-hole 可拦截 20%-30% 的 DNS 请求，网页加载速度提升 15% 以上（基于 SpeedTest 对比）。延伸推荐将树莓派扩展为家庭自动化中枢，例如通过 Home Assistant 实现设备联动。Pi-hole 的定期维护可通过 `pihole -g` 更新过滤列表，配合 crontab 设置每日自动任务：

```
0 3 * * * pihole updateGravity >/dev/null 2>&1
```

该技术方案以低于 5W 的功耗实现全年无间断守护，构建隐私与效率并重的家庭网络环境。