WisDefend Linux 部署手册

v1.9.3 | 更新于 2026-11-15
离线查阅版；复制命令请优先使用发布包内 DEPLOYMENT.md。

本文面向负责网站部署和安全运维的管理员，说明如何在 Linux 服务器上安装、配置、验证、升级和回滚 WisDefend。

本次手册更新对应 v1.9.3（2026-11-15）。升级前请先阅读第 10 节的配置兼容检查。

WisDefend 部署在公网访问入口与网站后端之间。外部请求先到 WisDefend，再由 WisDefend 转发到后端服务。

```
访客 / CDN -> WisDefend（80/443） -> 网站后端（例如 127.0.0.1:8080）
```

> WisDefend 只能清洗已经到达服务器的流量。超过服务器入口带宽的容量型攻击，需要配合运营商高防或 CDN。

1. 部署前检查

1.1 系统要求

项目	最低要求	建议
操作系统	Linux	使用仍在维护期内的主流发行版
CPU / 内存	1 核 / 1 GB	2 核及以上
架构	amd64 或 arm64	与下载包架构一致
运行方式	systemd	使用发布包附带的服务文件
网络	能接收站点流量	使用自动证书时需能访问 ACME 服务
XDP（可选）	Linux 5.4 及以上	内核和网卡驱动支持越完整越好；不可用时自动降级到 nftables

发布二进制不需要安装 Go。自行从源码构建时，需要 Go 1.27.4 或更新的受支持安全版本。 不要降低 go.mod 的版本要求；网关直接处理 HTTP/TLS，规则库不能替代标准库安全补丁。 升级工具链后，应重新执行单测、race、vet、漏洞扫描及目标 Linux 的 CDN/直连验收。

查看服务器架构：

```
uname -m
```

- 输出 x86_64：选择 linux-amd64 包。

- 输出 aarch64 或 arm64：选择 linux-arm64 包。

1.2 确认网络与端口

默认需要以下端口：

端口	用途	建议
TCP 80	HTTP、HTTPS 跳转、ACME 验证	对公网开放
TCP 443	HTTPS 网站流量	对公网开放
TCP 22	SSH 管理	只向可信来源开放
TCP 9527	管理控制台	默认只监听本机，不直接开放公网

确认 80 和 443 端口没有被 Nginx、Apache 或其他程序占用：

```
sudo ss -lntp | grep -E ':(80|443)\b' || true
```

如果已有反向代理占用 80/443，可以选择以下一种方式：

- 让 WisDefend 接管公网 80/443，并把原代理改到本机其他端口。

- 让现有代理继续位于最外层，把流量转发给 WisDefend 的自定义监听端口。

第二种方式必须正确配置真实客户端 IP。不要在未经信任的公网入口上直接信任 X-Forwarded-For。

还需要在云平台安全组和 Linux 防火墙中放行网站端口。以下命令按实际防火墙选择一组执行。

使用 firewalld：

```
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=https
sudo firewall-cmd --reload
```

使用 UFW：

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
```

不要因为部署方便而把 SSH 或管理控制台端口向所有来源开放。

1.3 准备站点信息

部署前记录以下信息：

- 需要保护的域名，例如 example.com 和 www.example.com。

- 后端地址，例如 http://127.0.0.1:8080。

- 是否由 WisDefend 自动申请 HTTPS 证书。

- 是否使用 CDN 或上游负载均衡。

- SSH 管理端口；如果不是 22，需要写入 defense.enforce.admin_ports。

2. 安装 WisDefend

以下示例假设已经把发布压缩包上传到服务器当前目录。

官方下载页：<https://wisdefend.xinruihuirong.com/download.html>

本手册 PDF：<https://wisdefend.xinruihuirong.com/dist/WisDefend-Linux-部署手册.pdf>

2.1 校验下载文件

如果发布包同时提供 .sha256 文件，先验证完整性：

```
sha256sum -c wisdefend-VERSION-linux-ARCH.tar.gz.sha256
```

把 VERSION 和 ARCH 替换成实际文件名中的版本与架构。

2.2 解压并安装

```
tar -xzf wisdefend-VERSION-linux-ARCH.tar.gz
cd wisdefend-VERSION-linux-ARCH

sudo mkdir -p /opt/wisdefend
sudo install -m 0755 wisdefend /opt/wisdefend/wisdefend
sudo install -m 0644 wisdefend.yaml /opt/wisdefend/wisdefend.yaml
sudo install -m 0644 wisdefend.service /etc/systemd/system/wisdefend.service
```

如果压缩包内包含 geoip.dat，一并复制：

```
sudo install -m 0644 geoip.dat /opt/wisdefend/geoip.dat
```

创建运行数据目录：

```
sudo mkdir -p /opt/wisdefend/data
sudo chmod 0700 /opt/wisdefend/data
```

3. 配置站点

编辑配置文件：

```
sudo vi /opt/wisdefend/wisdefend.yaml
```

3.1 最小 HTTPS 配置

下面的配置让 WisDefend 监听 80/443，自动申请证书，并把请求转发到本机 8080 端口：

```
server:
  http_addr: ":80"
  https_addr: ":443"

tls:
  acme:
    enabled: true
    email: "admin@example.com"
    staging: false
  storage_dir: "./data/certs"

defense:
  level: 2
  data_dir: "./data"

sites:
  - domains: ["example.com", "www.example.com"]
    upstream: "http://127.0.0.1:8080"
    tls: true
    force_https: true
```

首次上线建议从 2 级开始。确认业务正常后，再按需要提高防御级别。

3.2 仅使用 HTTP

内网站点或测试环境可以关闭 TLS：

```
tls:
  acme:
    enabled: false

sites:
  - domains: ["internal.example.com"]
    upstream: "http://127.0.0.1:8080"
    tls: false
```

3.3 HTTPS 后端

后端本身使用 HTTPS 时，将 upstream 写为 https:// ：

```
sites:
  - domains: ["secure.example.com"]
    upstream: "https://10.0.0.3:8443"
    tls: true
    force_https: true
```

只有后端使用无法验证的自签证书时，才设置：

```
    insecure_skip_verify: true
```

该选项会跳过后端证书校验，不建议用于正常生产环境。

3.4 CDN 或负载均衡场景

WisDefend 只会采信来自可信连接对端的访客 IP 请求头。不要只填写头名而不限制回源网段。

管理台入口：配置管理 → CDN 与接入。

- 接入方式选择“直连公网”或“CDN / 反向代理”。这不是自动识别开关，也不按国内、海外区分。

- CDN 接入的配置模板可选“自定义（通用 CDN / 反向代理）”、Cloudflare 公网或 Tunnel。 国内 CDN、自建反代及负载均衡使用自定义模板，并填写供应商确认的可信回源网段和来源头。

- “直连公网”对应 trusted_proxy 的 preset、cidrs、header 全部为空。 切换前需确认清空；更换模板会清除旧网段并重置头名，防止混用不同 CDN 的信任配置。 修改只在保存并校验通过后生效；“放弃修改”可恢复原配置。

- 旧版的“不使用预设”不一定是直连：已有网段或头名时，新表单会回显为 CDN / 反向代理的自定义模板。 没有新增 YAML 字段，旧配置可继续使用。

表单不会自动关闭回源保护、修改 XDP 或添加访客豁免。切换到直连后，请核对“内核处置”中的 回源保护；若仍依赖 CDN 允许列表，公网访问仍会被阻断，保存也可能因缺少允许网段被拒绝。 当前只有一组全局来源头规则，不支持按多个 CDN 分别选择头名。不要在 Cloudflare 预设中 直接追加使用其他来源头的厂商节点。模板不是厂商完整兼容性认证，接入后仍需验证真实 IP、 Host/SNI、浏览器挑战、Cookie 与 CDN 缓存策略；不会自动下载国内 CDN 网段。

Cloudflare 公网代理可直接使用内置预设：

```
defense:
  trusted_proxy:
    preset: cloudflare
    cidrs: []
    header: ""
```

Cloudflare Tunnel 在本机运行 cloudflared 时使用：

```
defense:
  trusted_proxy:
    preset: cloudflare_tunnel
    cidrs: []
    header: ""
```

其他 CDN 或负载均衡使用自定义网段。以下示例表示只有 10.20.0.0/16 的连接才能提供 X-Forwarded-For：

```
defense:
  trusted_proxy:
    preset: ""
    cidrs: ["10.20.0.0/16"]
    header: X-Forwarded-For
```

cidrs 禁止填写 0.0.0.0/0 或 ::/0。直连请求携带的来源头会被忽略并从回源请求中删除。 可信代理缺少来源头、头值非法或 X-Forwarded-For 链畸形时，WisDefend 返回 400，不回退到代理节点 IP。 例外：环回对端完全不带来源头时按直连处理，供 Tunnel 同机健康检查使用；存在但畸形的头仍返回 400。

Cloudflare 注意事项：

- 不要启用“移除访客 IP 头”托管转换；

- Pseudo IPv4 必须使用“添加头”模式。覆盖模式会被拒绝并返回 400；

- 内置网段是静态快照。升级 WisDefend 或在 cidrs 中追加 Cloudflare 新网段；

- Cloudflare 公网网段不能证明请求属于你的网站。需要更严格的回源隔离时优先使用 Tunnel；

- 配置可信代理后，不建议启用 XDP 按源 IP 限速。XDP 看到的是 CDN 节点，可能误伤整批访客。

首次接入 Cloudflare 并使用 ACME 自动证书时，按以下顺序操作：

1. DNS 先设为“仅 DNS”，让 WisDefend 完成首次证书签发。

2. Cloudflare SSL/TLS 模式设为 Full (strict)。

3. 开启代理，确认 HTTPS 与续期路径都正常。

80 与 443 监听都能应答 ACME HTTP-01。不要在 Cloudflare 侧阻断 /.well-known/acme-challenge/。

如需阻止访客绕过 CDN 直连源站，可同时启用回源保护：

```
defense:
  enforce:
    enabled: true
    origin_guard:
      enabled: true
      use_trusted_proxy: true
      allow_v4: []
      allow_v6: []
```

回源保护按 IPv4、IPv6 分别生效。某个地址族没有允许网段时，该地址族会被完全阻断。 上线前必须执行 -print-rules，并确认 SSH 管理端口位于 admin_ports。

3.5 API、App 和回调接口

4 级防御会要求新访客完成浏览器验证，普通 API 客户端无法完成该流程。应提前配置豁免路径：

```
defense:
  allowlist:
    api_paths: ["/api/", "/webhook/"]
```

只添加确实需要自动访问的最小路径范围。

API 豁免只匹配无歧义的规范路径，并按完整路径段匹配。例如 /hook 不会豁免 /hook-other。含 .、..、重复斜杠、分号、反斜杠或歧义编码的请求不会取得 API 豁免，而是继续走正常防御管线；程序不会先清洗路径再授予豁免。

挑战验证允许因响应丢失而重试。同一张票据始终复用同一个计量 ID，不因重试、 修改主动指纹或 nonce 重置频率和行为记录。新票据使用更长的随机数；升级前两分钟内 尚未完成的旧格式挑战票据仍可验证，也遵循同一计量规则，无需额外消费记录表。

4. 配置管理控制台

控制台可以修改防御行为，默认关闭且只建议监听本机。

生成随机密码：

```
openssl rand -hex 16
```

配置示例：

```
admin:
  enabled: true
  addr: "127.0.0.1:9527"
```

```
  username: "wafadmin"
  password: "替换为随机生成的密码"
  session_ttl: "12h"
  secure_cookie: false
  geoip_file: "./geoip.dat"
```

从管理员电脑通过 SSH 隧道访问：

```
ssh -L 9527:127.0.0.1:9527 USER@SERVER_IP
```

保持 SSH 会话打开，然后在本机浏览器访问：

```
http://127.0.0.1:9527/
```

不建议把 9527 端口直接暴露在公网。如果必须远程开放，应在外层增加访问控制和额外认证。

如果由 Nginx、Cloudflare 等代理终止 HTTPS，再通过 HTTP 连接控制台，设置 admin.secure_cookie: true 并重启。登录和退出 Cookie 都会携带 Secure，浏览器只在 HTTPS 上发送会话。程序不根据 X-Forwarded-Proto 自动开启此项，以免信任伪造的头。 本机 HTTP 或上述 SSH 隧道用法保持 false，否则部分浏览器将无法保持登录。

管理台写操作使用与登录会话绑定的 CSRF 令牌，并拒绝浏览器标记为跨站或兄弟子域 来源的写请求。网页会自动携带令牌，不需要额外配置，也不依赖代理转发的协议头。 反向代理不要为管理台放宽跨源访问权限。

如果使用运维脚本调用接口，先以 application/json 登录并保留会话 Cookie，再以 该 Cookie 请求 GET /api/csrf，取响应中的 csrf_token。调级、保存配置、安装授权、 上传规则及退出时，均须带 X-CSRF-Token。例如：

```
POST /api/level HTTP/1.1
Cookie: wd_admin_session=替换为当前会话
X-CSRF-Token: 替换为该会话的csrf_token
Content-Type: application/json

{"level":3}
```

退出接口只接受 POST。会话过期或进程重启后，应重新登录并取得新令牌；不能用另一 会话的令牌代替。授权失效不影响已有会话获取退出令牌并安全退出。

5. 安装离线授权

无需开启 Web 控制台即可购买、安装或续购授权。先在客户服务器上查看授权说明：

```
cd /opt/wisdefend
./wisdefend -license-help
```

用自己的电脑或手机打开 自助授权中心， 完成邮箱验证，填写服务器指纹，选择套餐并付款。命令会显示指纹；也可单独获取：

```
./wisdefend -machine-id
```

-machine-id 的标准输出只有一行指纹，自助授权地址与帮助提示写入标准错误， 不影响脚本读取指纹。-license-help 无需配置文件，不启动网关，也不发起网络请求。 旧版若不识别 -license-help，可直接访问上述地址；网站签发的新密钥授权需要兼容的网关版本， 旧公开版的兼容边界见 [授权说明](license.md#兼容密钥与在线专用密钥共存)。

请保存订单号和查询密钥。签发成功后，可在订单中领取授权码或文件，系统也会发送到邮箱。 收到后按以下方式安装；也可联系技术支持协助签发。

5.1 通过控制台安装

登录控制台，打开 关于与授权。授权输入框旁的 自助授权 按钮可打开授权中心。 取得授权后，粘贴以 WD1- 开头的授权码并提交。

控制台服务可按配置启动，但每次请求都会重新检查 admin 功能授权。授权缺失、过期或 不包含 admin 时，已有会话也会收到 403；安装有效授权后可恢复访问，无需重新创建服务。 首次授权无法进入控制台时，应使用文件方式。

5.2 通过文件安装

将收到的 wisdefend-license.key 文件或完整 WD1- 授权码保存为 /opt/wisdefend/wisdefend-license.key，无需改名。 不要把整封邮件或订单号写入文件。限制文件权限，并确保服务运行用户可读：

```
sudo chmod 0600 /opt/wisdefend/wisdefend-license.key
```

确认配置为：

```
license:
  file: "./wisdefend-license.key"
  expired_mode: basic
  bind_machine: true
```

以上是新配置的默认路径。已有配置若明确指定 license.json 或其他名称，继续按该路径安装； 程序不会自动重命名或迁移授权文件。如果旧配置未指定 license.file，而文件仍叫 license.json，升级前请显式填写旧路径。

运行中的网关每 5 分钟复查一次，替换文件后无需开启控制台即可自动恢复授权。 如需立即重新校验，可重启服务（重启可能短暂影响现有连接）：

```
sudo systemctl restart wisdefend
```

授权缺失、无效或过期不会中断网站代理。默认 basic 模式会把最高防御能力降到 2 级。 已加载的 XDP 也受 xdp 功能授权约束：授权失效或功能被移除时会动态卸载；授权恢复后 会重新加载，并补写当前封禁名单。

若卸载失败，后端进入“清理未完成”状态，不再报告为可用，也不会假装恢复成功。 后续授权复查会重试；重新启用前必须先清理旧句柄，再创建新后端并补发黑名单。 此时七层防护和仍可用的 nftables 继续工作。日志中的“XDP 卸载不完整”需要排查， 不能仅凭“授权有效”认定 XDP 已恢复。

5.3 续购与日志提示

用订单号和查询密钥查询原订单，通过原订单的 续购 入口操作。 续购后请安装新授权文件；付款本身不会改动服务器上的旧文件。

授权缺失、无效或到期时，状态变化日志会提示自助授权地址及 wisdefend -license-help。 系统服务日志可用以下命令查看：

```
sudo journalctl -u wisdefend -n 100 --no-pager
```

这些入口只提供人工访问指引，网关仍完全离线验签，不会自动上传机器指纹或访问订单中心。

5.4 无授权时验证基础防护

正式发布包在缺少授权文件时仍可反向代理。默认 license.expired_mode: basic 将生效防御级别限制在最高 2 级；管理控制台会返回 403，XDP 不可用。若显式配置 passthrough，则只保留 1 级代理。此处的“免费”指正式发布包的无授权状态，不适用于未编入授权公钥的开发构建。

先用 sudo journalctl -u wisdefend -n 100 --no-pager 查看启动日志。正式包缺少授权文件时应提示“授权无效，防御能力已降级”；如果配置级别高于 2，还应看到“防御级别受授权限制”，并显示实际生效级别为 2。若出现“未配置授权公钥，按不限制模式运行”，说明当前二进制未启用授权校验，应停止使用该包并联系发布方。

免费档没有态势控制台。要验证七层拦截，在不含真实敏感文件的隔离测试站点上，确认 defense.waf.enabled: true 且 defense.waf.mode: block，再从网关服务器向其 HTTPS 入口发送一条受控请求：

```
SITE_DOMAIN=your-test-domain.example
curl -i --resolve "${SITE_DOMAIN}:443:127.0.0.1" "https://${SITE_DOMAIN}/.env"
```

内置规则 WD-6001 匹配该路径，预期由网关返回 429 Too Many Requests；同时核对测试后端访问日志，确认请求未回源。若需要查看判定原因，可临时将 log.level 设为 debug 并热加载，再查看 journalctl -u wisdefend -n 100 --no-pager 中的“拒绝请求”和 signature:WD-6001；验证后改回 info，避免长期记录大量访问路径。正常请求还应能够通过代理到达后端。

上述请求验证的是特征库拦截，不等同于验证 2 级频率清洗。频率清洗应在隔离环境用受控流量、按实际阈值另行检查 429、拒绝原因及后端未收到被拒请求。-print-rules 只打印拟安装的 nftables 规则，不能证明七层拦截或授权降级已经生效。

6. 启动前检查

6.1 检查后端

先从 WisDefend 服务器直接访问后端：

```
curl -I http://127.0.0.1:8080/
```

如果这里失败，应先修复后端监听地址、防火墙或应用服务。

6.2 检查内核规则

启用 defense.enforce 前，先打印将要安装的 nftables 规则：

```
cd /opt/wisdefend
sudo ./wisdefend -c ./wisdefend.yaml -print-rules
```

首次启用内核处置时，建议先配置：

```
defense:
  enforce:
    enabled: true
    dry_run: true
    admin_ports: [22]
```

如果 SSH 使用其他端口，例如 2222，应改为：

```
    admin_ports: [2222]
```

确认日志和规则符合预期后，再把 dry_run 改为 false。

启用 defense.enforce.synproxy.enabled 前，还需确认整机的连接跟踪和 TCP 参数：

```
sysctl net.netfilter.nf_conntrack_tcp_loose net.ipv4.tcp_syncookies net.ipv4.tcp_timestamps
```

三项预期分别为 0、1、1。程序只检查，不自动修改这些全局参数。 参数缺失、值不符或内核不支持 SYNPROXY 时，仅关闭 SYNPROXY 并记录原因； 黑名单、UDP 策略及回源保护仍可生效。修改 nf_conntrack_tcp_loose 会影响整机的 连接跟踪，请先在预发布环境验证其他服务。-print-rules 只打印配置规则，不代表运行期 前置条件已通过；以启动日志和 nft list table inet wisdefend 为准。

SYNPROXY 只保护直接交付本机的业务端口。notrack 使用 FIB 路由查询确认目标为本机， 并排除环回和管理端口，不取消普通 Docker/VPN 出站转发连接的连接跟踪。 程序同时探测本机目标筛选表达式；该表达式不可用时也只关闭 SYNPROXY。 不要让 SYNPROXY 与 Docker 端口发布、DNAT 共用同一入口 IP/端口；这类入口应先调整 网络拓扑或关闭 SYNPROXY，不能把本机服务的 SYN 代答规则直接用于 NAT 后端。

升级程序并重启后，可以检查实际安装的作用域：

```
sudo nft list chain inet wisdefend prerouting
```

启用成功时，notrack 所在行应包含 iifname != "lo"、fib daddr type local，以及 配置了管理端口时的 tcp dport != @admin_ports。SYNPROXY 规则不通过 SIGHUP 热更新， 升级或修改该项后需要重启。还应从业务容器内验证其实际使用的外部 HTTP/HTTPS 服务。

控制台统计接口的 xdp_rate_state 区分 active（限速启用）、degraded（原子限速 不可用，XDP 黑名单和 UDP 防护仍启用）、unavailable（配置了 XDP 但当前未挂载） 及 disabled（未配置此限速或 XDP）。降级状态也会显示在仪表盘。 每次运行期授权复查都会重试降级的原子限速；失败不卸载基础防护，成功时保留已有 黑名单和统计。复查默认每 5 分钟执行，不需要通过切换许可证强制重载。

XDP、SYNPROXY、UDP 丢弃策略、sysctl 调优和回源保护都属于内核处置子功能， 必须同时设置 defense.enforce.enabled: true。父开关关闭但子功能启用时， 配置会被拒绝，不会静默忽略这些设置。

6.3 前台试运行

正式注册 systemd 服务前，可以前台启动：

```
cd /opt/wisdefend
sudo ./wisdefend -c ./wisdefend.yaml
```

确认没有配置错误后，按 Ctrl+C 停止。不要在 systemd 服务运行时重复前台启动，否则会因端口占用而失败。

正常停止时，程序会等待内核处置结果回写，并停止、等待周期统计保存，最后再写入 最新的加密仪表盘快照 data/dashboard.dat。快照使用独立的 0600 临时文件，文件同步后 原子替换，并同步所在目录，避免旧周期快照覆盖退出时的新状态。 SIGKILL、断电等异常退出无法执行最终保存，统计仍可能回退到最近一次周期快照。

7. 注册并启动 systemd 服务

```
sudo systemctl daemon-reload
sudo systemctl enable --now wisdefend
```

查看状态：

```
sudo systemctl status wisdefend --no-pager
```

持续查看日志：

```
sudo journalctl -u wisdefend -f
```

服务文件默认设置：

- 工作目录：/opt/wisdefend

- 配置文件：/opt/wisdefend/wisdefend.yaml

- 异常退出后自动重启

- 文件描述符上限：1,048,576

- 具备监听低端口及管理 nftables、XDP 所需的系统能力

启用 XDP 后可检查网卡挂载状态：

```
ip -details link show
```

输出中出现 xdp 或 prog/xdp 表示程序已挂载。XDP 是否能以驱动模式运行取决于网卡 和驱动；不支持时 WisDefend 会尝试通用模式，仍不可用则保留 nftables 防护并记录日志。

8. DNS 切换与上线验证

8.1 切换前

建议提前降低域名 DNS TTL，例如降到 300 秒，并保存原解析地址以便回滚。

如果使用自动 HTTPS 证书，域名的 A/AAAA 记录必须指向 WisDefend，且公网能够访问 80/443 端口。

8.2 修改 DNS

把域名的 A 记录改为 WisDefend 服务器公网 IPv4 地址。只有服务器具备可用公网 IPv6 时才配置 AAAA 记录。

检查解析：

```
dig +short example.com A
```

```
dig +short example.com AAAA
```

8.3 验证网站

```
curl -I http://example.com/
curl -I https://example.com/
```

重点确认：

- HTTP 是否按配置跳转 HTTPS。

- HTTPS 证书域名与有效期是否正确。

- 页面、登录、上传、下载和 API 是否正常。

- 后端日志中是否能看到请求。

- WisDefend 日志中是否出现大量误拦截。

上线初期可以把 WAF 调整为只记录：

```
defense:
  waf:
    enabled: true
    mode: "log"
```

观察正常业务后，再改为 block。

9. 日常运维

9.1 修改配置并热加载

先备份配置：

```
sudo cp /opt/wisdefend/wisdefend.yaml /opt/wisdefend/wisdefend.yaml.bak
sudo vi /opt/wisdefend/wisdefend.yaml
sudo systemctl reload wisdefend
```

查看热加载结果：

```
sudo journalctl -u wisdefend -n 100 --no-pager
```

支持热加载的配置包括防御级别、站点、频率阈值、特征库、访客验证页参数、 吊销容量与有效期、豁免规则、缓存和日志级别。

以下配置需要重启：监听地址、ACME/证书存储设置、资源档位、数据目录、 信任库活跃容量与周期清理间隔、控制台与离线授权、内核处置配置。 其中 data_dir 改动前，应先把旧目录数据完整迁移到新目录。

微缓存同时识别 max-age=1 和 max-age="1"，共享缓存优先使用 s-maxage。 有效期为零、指令重复冲突或参数无法可靠解析时不会缓存，不会回退成配置中的较长 TTL。 仅使用 Expires 时，缓存按 Expires - Date 计算新鲜期，再扣除响应年龄；Age 已超过 新鲜期的响应不会因为 Expires 在本地时钟的未来而被重新缓存。 响应年龄也包含回源等待、正文传输及本地驻留时间。例如 max-age=60、Age=59， 回源再耗时两秒的响应已过期，不会供下一次请求命中。命中返回的 Age 同样使用 修正后的年龄，不会从写入本地缓存时重新计时。

HTTP/1.1 与 HTTP/2 的无正文 GET 都可以进入微缓存。HTTP/2 请求不能仅凭 Body 对象是否存在判断；有正文、未知长度，或显式声明零长度但不能确认流已结束的请求 仍保守绕过缓存。Cookie、Authorization、Range、条件请求及请求侧缓存指令同样绕过。 Cloudflare 会独立处理缓存和 Range，排查时同时检查 CF-Cache-Status、网关 X-Cache 及回源计数，不要把边缘命中当作网关缓存命中。

```
sudo systemctl restart wisdefend
```

9.2 常用命令

```
# 查看运行状态
sudo systemctl status wisdefend --no-pager

# 查看最近 200 行日志
sudo journalctl -u wisdefend -n 200 --no-pager

# 热加载配置
sudo systemctl reload wisdefend

# 重启服务
sudo systemctl restart wisdefend

# 查看版本
/opt/wisdefend/wisdefend -version

# 查看机器指纹
/opt/wisdefend/wisdefend -machine-id
```

9.3 需要备份的文件

建议定期备份：

```
/opt/wisdefend/wisdefend.yaml
/opt/wisdefend/wisdefend-license.key
/opt/wisdefend/data/
/opt/wisdefend/geoip.dat（如有）
```

data/ 包含证书、挑战通行证密钥、吊销状态和旧证迁移记录。丢失后可能需要重新申请证书， 已验证访客也会失去原有信任状态。 授权文件的备份路径以实际 license.file 配置为准；旧配置可能仍使用 license.json。

10. 升级与回滚

升级前准备新二进制，并确认架构正确。

10.1 v1.9.3 配置兼容检查

先使用新二进制检查现有配置；此命令只校验和生成规则，不安装内核规则，也不启动监听：

```
./wisdefend -c /opt/wisdefend/wisdefend.yaml -print-rules > /dev/null
```

若内核处置未启用，此命令会在完成配置加载校验后提示 defense.enforce.enabled 为 false，不会安装任何防火墙规则，并以非零状态退出。 这是预期提示，不必为校验而打开父开关。其他配置错误应先修正，再替换运行中的程序。 本版新增或加强以下检查：

- API 豁免不能包含 /、空值或非规范路径。此类配置会阻止启动或保存，不能用作全站放行开关。

- 启用 XDP、SYNPROXY、UDP 丢弃、sysctl 调优或回源保护时，必须打开内核处置父开关。 不需要内核能力时，应关闭相应子功能，而不是仅关闭父开关。

- SYNPROXY 的 MSS 必须为 1..65535，wscale 必须为 0..14。

- XDP 的包速率和突发值不能超过 4294967295；管理端口最多 64 个，UDP 放行端口最多 1024 个。

本版不改变通行证格式、授权文件路径或特征库版本。保留配置、授权文件和完整数据目录， 不要为了升级清空信任库。API 豁免及缓存的开启、关闭支持控制台保存和 SIGHUP 热加载。

10.2 历史版本兼容与数据备份

v1.9.0 新增可信前置代理 / CDN 模式，并把通行证升级为 v4。升级前必须备份配置、 授权文件和完整数据目录。旧配置中的非空 defense.real_ip_header 会被拒绝；请改为 defense.trusted_proxy。旧字段为空时可直接删除。

> 通行证格式升级提示：必须保留 /opt/wisdefend/data/。新版会把有效的 v1/v2/v3 通行证引导到一次性迁移挑战，成功后换发 v4。即使升级时处于 5 级，持有 未过期、未吊销旧证的用户仍可迁移，新证会保留原签发时间；已经过期、已吊销或 无法验签的旧证不能借迁移绕过准入。如数据目录、信任库或通行证密钥丢失，应先 降到 4 级让正常用户重新验证。

吊销状态现在有独立容量与有效期。任一吊销表达到 80% 时，系统停止签发新证和迁移旧证； 回落到 70% 以下后恢复。表满或存储写入失败时，控制台会显示紧急或存储故障状态； 存储故障期间停止签发和迁移；已有 v4 按内存吊销记录裁决，不会统一拒绝所有持证用户。 已接受的吊销继续在内存生效，等待存储恢复后补写；补写前异常重启可能丢失这些未落盘记录。 challenge.revoke_max、revoke_ttl 与 token_ttl 支持热加载；可先调大容量并保存配置， 确认占用回落和存储恢复。只有无法修复数据时才考虑重置信任库并轮换密钥。 调大 token_ttl 只影响此后签发的通行证，不能使过期旧证复活；调小立即生效。 有效期调整历史保存在数据库内，保存失败时拒绝发布新值。请同时备份完整数据库。 旧证命中吊销只拒绝，不再生成新身份吊销记录，避免通过改变指纹填满吊销表。

控制台的信任状态提示会分别显示活跃记录、主体/再准入/旧身份吊销记录、迁移记录、 容量占用、失败次数与估算内存。估算值包含 Go 映射和记录的近似开销，只用于容量规划， 不等同于进程 RSS。

吊销表只回收到期记录，不会为腾空间提前放行仍被吊销的身份。 后台每 5 秒读取两张表的条数，分别计算最近最多 1 分钟的净增长。 在停发之前，任一表达到 70%，或按当前趋势预计 15 分钟内达到 80%，即显示 吊销容量预警。 鼠标悬停在 工作状态 上可查看净增长和预计剩余时间。日志事件名为 revocation_capacity_warning，持续预警时最多每 5 分钟记录一次；关闭管理台也会采样和记录。

例如，容量 50,000 的表已占 32,000 条，若净增 1,000 条/分钟，预计 8 分钟达到停发线， 此时占用虽不到 70%，也会预警。预测不保证预留时间，突发增长仍可能直接触及停发线。 收到预警后，先排查异常吊销来源，按需加强 CDN 边缘过滤；若内存有余量，再考虑调大 revoke_max。

信任数据损坏时服务会拒绝启动，不会覆盖原文件。确认已备份完整数据库后，可在防御级别 低于 5 时执行：

```
sudo systemctl stop wisdefend
cd /opt/wisdefend
sudo ./wisdefend -c ./wisdefend.yaml -reset-trust
sudo systemctl start wisdefend
```

命令先创建权限为 0600 的完整数据库备份，再清空信任状态并默认轮换通行证密钥。 -keep-token-key 会让此前被吊销的通行证重新有效，仅限明确接受该风险时使用。

经 CDN 签发的 v4 验签绑定浏览器 UA 和语言，不绑定 CDN 回源 TLS 指纹；主动指纹只参与吊销主体派生。 被盗旧证可能被他人抢先迁移；原持有者需要在 4 级重新验证。该风险窗口最长为一个 challenge.token_ttl。

10.3 备份当前版本

先停服务，再复制完整数据目录，避免获得不一致的 bbolt 数据库副本。 备份完成后可先启动旧版，安装新版时再短暂停机。

```
sudo systemctl stop wisdefend
sudo cp /opt/wisdefend/wisdefend /opt/wisdefend/wisdefend.previous
sudo cp /opt/wisdefend/wisdefend.yaml /opt/wisdefend/wisdefend.yaml.previous
sudo cp -a /opt/wisdefend/data /opt/wisdefend/data.previous
sudo systemctl start wisdefend
```

不要删除 /opt/wisdefend/data/ 和授权文件。

10.4 安装新版本

在新版本解压目录执行：

```
./wisdefend -version
sudo systemctl stop wisdefend
sudo install -m 0755 ./wisdefend /opt/wisdefend/wisdefend
sudo systemctl start wisdefend
sudo systemctl status wisdefend --no-pager
```

然后检查日志和业务：

```
sudo journalctl -u wisdefend -n 100 --no-pager
curl -I https://example.com/
```

建议再完成以下检查：

```
# 查看实际运行版本
/opt/wisdefend/wisdefend -version

# 使用 XDP 时确认挂载状态
ip -details link show
```

若正在使用 4/5 级防御，请用真实浏览器访问一次：有效旧证应完成一次迁移后正常进入， 已过期、已吊销或无法验签的旧证不得获得新通行证。1 ~ 3 级不会因旧 Cookie 被强制拉入 迁移挑战。

二进制升级需要重启服务，会有短暂连接中断。建议在低峰期操作。

10.5 回滚

v1.8.7 及更早版本不认识 v1.9.0 的 v4 通行证、独立吊销表和迁移消费表。只替换旧二进制会 丢失新版信任状态，不能作为无损降级。需要保留升级前的完整数据库备份；若必须回滚， 应在停机状态下一并恢复当时的二进制、配置和完整数据目录。

新版本异常时执行：

```
sudo systemctl stop wisdefend
sudo cp /opt/wisdefend/wisdefend.previous /opt/wisdefend/wisdefend
sudo cp /opt/wisdefend/wisdefend.yaml.previous /opt/wisdefend/wisdefend.yaml
sudo mv /opt/wisdefend/data /opt/wisdefend/data.v1.9.0.failed
sudo cp -a /opt/wisdefend/data.previous /opt/wisdefend/data
sudo chmod 0755 /opt/wisdefend/wisdefend
sudo systemctl start wisdefend
```

回滚后再次检查状态、日志和网站访问。

11. 故障排查

11.1 服务启动失败

查看详细错误：

```
sudo systemctl status wisdefend --no-pager
sudo journalctl -u wisdefend -n 200 --no-pager
```

常见原因：

- YAML 缩进或字段值错误。

- 80/443 已被其他程序占用。

- 后端地址格式错误。

- 数据目录没有写权限。

- 启用了内核处置，但系统缺少相应能力。

11.2 提示端口被占用

```
sudo ss -lntp | grep -E ':(80|443|9527)\b'
```

停止冲突服务，或修改 WisDefend 的监听地址。不要让两个服务同时监听同一地址和端口。

11.3 出现 502 或后端不可达

从 WisDefend 服务器直接测试后端：

```
curl -v http://127.0.0.1:8080/
```

检查：

- 后端进程是否运行。

- sites[].upstream 的协议、地址和端口是否正确。

- 后端是否只监听了其他地址。

- 本机或后端防火墙是否拦截。

- HTTPS 后端证书是否可信。

11.4 自动证书申请失败

检查：

- 域名 A/AAAA 记录是否指向本机。

- 80/443 是否允许公网访问。

- 是否存在错误的 AAAA 记录。

- 服务器时间是否准确。

- 是否频繁测试触发 ACME 限额。

调试期间可以设置 tls.acme.staging: true。测试证书不受浏览器信任，正式上线前必须改回 false。

11.5 正常请求被拦截

先查看日志中的规则编号和拒绝原因。可临时将 WAF 调整为观察模式：

```
defense:
  waf:
    mode: "log"
```

确认某条规则误报后，可以只关闭该规则：

```
defense:
  waf:
    disabled_rules: ["WD-1004"]
```

不要为解决单个误报而关闭全部防护。大文件或二进制上传接口可以使用 skip_body_paths 跳过正文检查。

如果命中 WD-INPUT-LIMIT，表示请求超出可完整检查的长度：路径 4KB、查询串 8KB、 User-Agent 1KB、Referer 2KB、合并后的 Cookie 8KB。阻断模式会拒绝，而不是只检查 前缀后放行尾部；观察模式只记录，不阻断。正文上限仍由 waf.max_body_kb 单独控制， 不能靠调大该值放宽查询串或 Cookie。优先缩短请求或调整应用设计，不要为此扩大 API 豁免。

WAF 会按 Cookie 语义合并检查全部 Cookie 头。重复的 User-Agent 或 Referer 可能造成 后端解析差异，按 WD-HEADER-AMBIGUOUS 处理。

WAF 同时检查原始内容和解码视图：Cookie 做一次 URL 解码；JSON 还原字符串转义； XML 还原字符实体及 CDATA ，仍保留原始 XXE 声明检查。JSON/XML 也按正文形态识别， 不能仅靠伪装 Content-Type 跳过。声明为 HTML 的正文使用 HTML 解析，避免把正常 HTML 片段当作损坏的 XML。检查不会改变发给后端的正文。 HTML 以引号、花括号或方括号开头时，不会仅因 JSON 嗅探语法失败而拒绝； 但仍会检查已解码的 JSON 前缀，不能靠改成 HTML 类型并追加尾部来隐藏编码载荷。

multipart/form-data 必须提供非空 boundary，缺少时报告 WD-BODY-PARSE。 表单必须直接从 boundary 分隔行开始（可有前导空白行），不接受会被 MIME 解析器忽略的 非空前导区，避免其中藏有可被后端读取的 JSON 等正文。 每个文本字段分别检查原文、URL 解码内容和 JSON/XML/HTML 解码视图；字段名和文件名 独立检查，不把不同字段拼成一个结构化文档，也不检查文件内容。原始表单仍完整回源。 每份表单最多 1024 个 part（含文件）；文本字段的歧义媒体类型、压缩编码、嵌套 multipart 或超过解析资源上限时，报告 WD-BODY-PARSE。确需复杂上传协议时，请为专用上传路径 评估 skip_body_paths，不要扩大所有 API 的豁免范围。

文本字段的 Content-Transfer-Encoding 只允许空值、identity、7bit、8bit 或 binary。 base64、quoted-printable 及重复的该编码头会被拒绝；检查使用原始 part 头， 避免 MIME 解析器先解码并隐藏头部。带 filename 的 part 按文件处理，不检查其正文； 后端不得再把此类 part 当作普通文本参数解码执行。

结构化检查支持 UTF-8 / ASCII，并受资源上限约束：JSON/XML 深度 64、解析令牌数 16384（XML/HTML 属性也计数）；multipart 的每个文本字段分别受这些限制。 XML 单标签及 HTML 单令牌 64KB，解码输出不超过 原文两倍加少量分隔开销。损坏、无法支持的编码或超限正文按 WD-BODY-PARSE 处理； 观察模式仍只记录。正文缓冲预算按实际长度的八倍保守预留，总预算保持 128MB。

11.6 4 级下 App 或接口无法访问

TLS 指纹异常只作用于本次请求，不能因同一出口先前有过脚本访问就长期污染其他 浏览器的判断。严格准入仍使用 TLS、行为与浏览器挑战等启发式信号，不保证所有 浏览器和企业 TLS 中间设备均兼容；遇到异常先查看 automation:* 的具体原因。 不要仅靠更换 Cookie 判断问题已消失，也不要为此直接清空吊销库。

将必要的接口前缀加入：

```
defense:
  allowlist:
    api_paths: ["/api/", "/webhook/"]
```

保存后立即热加载，SIGHUP 同样生效。配置会拒绝 /、空路径、非规范路径， 以及包含 ?、#、%、;、反斜杠或控制字符的前缀。 API 豁免不参与频率、行为、IP/路径/通行证计量和拒绝率计算，但仍计入全局容量观测， 也仍接受 WAF 检查。豁免可绕过 5 级纯白名单准入；若只允许老用户访问， 请勿保留通用 API 豁免前缀。

/__wd/verify 有独立的固定哈希桶预限速：16384 个随机哈希桶，每桶每分钟最多 120 次。 不同来源可能共享同一桶；额度耗尽时返回 429 和 Retry-After: 60。 这不是所有网站请求的频率阈值，也不是可无限扩容的来源表。 请求正文读取完成后才检查挑战票据时效，慢速上传不能延长票据的两分钟有效期。

11.7 控制台无法访问

确认控制台已启用，并检查监听地址：

```
sudo ss -lntp | grep ':9527\b'
```

如果配置为 127.0.0.1:9527，远程浏览器不能直接访问，这是正常的。请使用 SSH 隧道。

11.8 授权无效

检查：

- 授权码是否完整复制。

- license.file 路径是否正确。

- 当前机器指纹是否与签发时一致。

- 授权是否过期。

- 程序是否为编入正确授权公钥的正式版本。

授权异常只会降低防御能力，不会中断网站代理。

12. 停用与卸载

停止并取消开机启动：

```
sudo systemctl disable --now wisdefend
```

确认网站流量已经切回其他入口后，再移除服务文件：

```
sudo rm /etc/systemd/system/wisdefend.service
sudo systemctl daemon-reload
```

建议先保留 /opt/wisdefend 作为可回滚备份。确认不再需要证书、配置和运行数据后，再由管理员按备份策略处理该目录。

13. 获取支持

联系技术支持时，请一并提供：

- WisDefend 版本：/opt/wisdefend/wisdefend -version
- Linux 发行版和内核：cat /etc/os-release && uname -r
- 最近日志：journalctl -u wisdefend -n 200 --no-pager
- 问题发生时间和操作步骤
- 已脱敏的相关配置

不要发送私钥、完整管理员密码、Cookie 或其他业务凭证。

联系方式：

- QQ：1940728253
- 微信：xwzy201130
- 邮箱：kefu@xinruihuirong.com
- 官网：<https://wisdefend.xinruihuirong.com/>