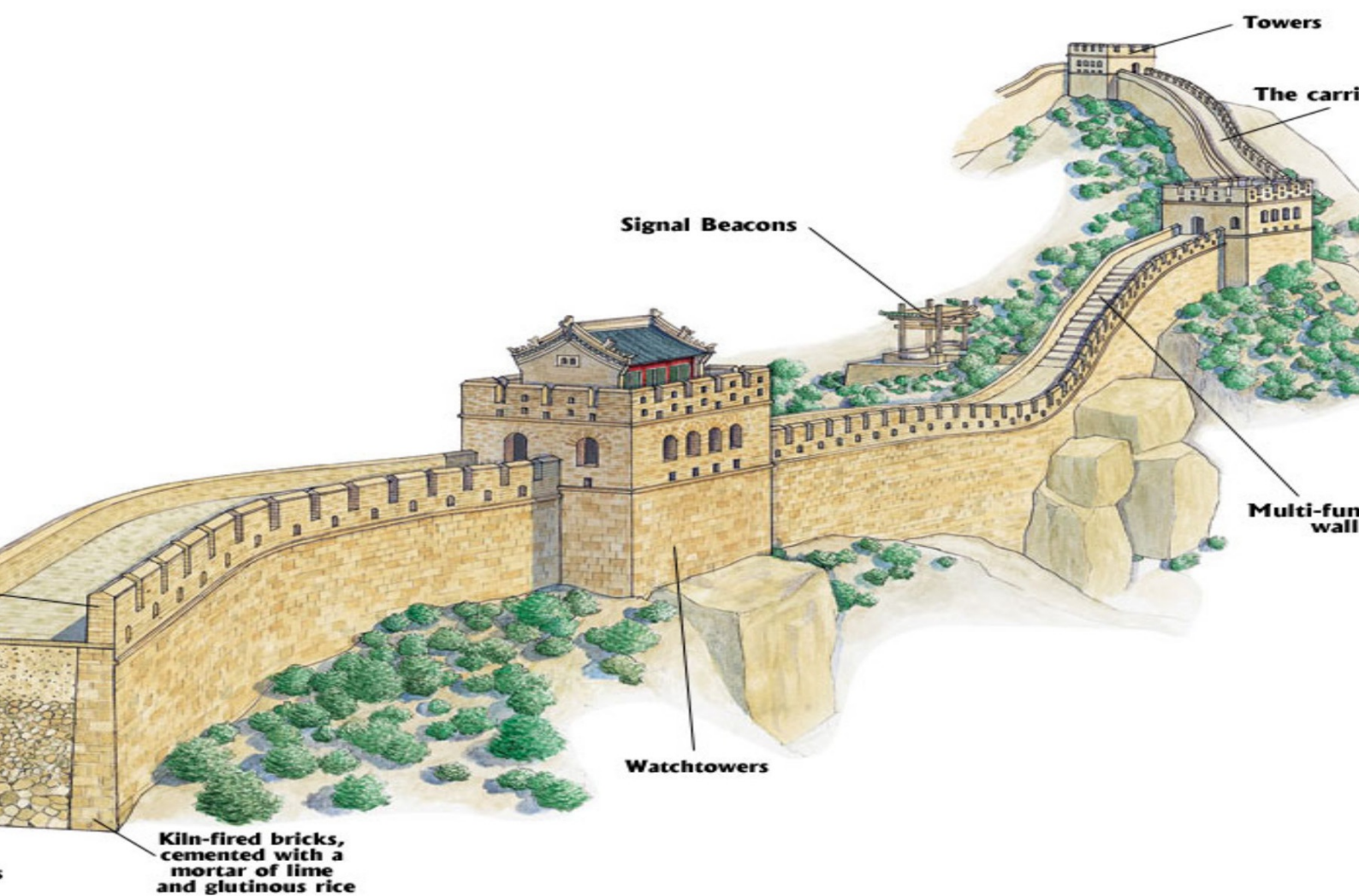


穿越长城



目录

Introduction	1.1
无线路由器刷OpenWrt固件的准备工作	1.2
什么是无线路由器固件	1.2.1
支持刷OpenWrt路由器列表推荐	1.2.2
备份原厂路由器配置文件	1.2.3
路由器怎样刷OpenWrt固件 (WR2543N为例)	1.3
怎样从官网下载OpenWrt固件	1.3.1
进管理页面刷OpenWrt教程	1.3.2
管理页面OpenWrt自动拨号上网设置教程	1.3.3
管理页面OpenWrt开启、设置无线(Wifi)教程	1.3.4
管理页面备份OpenWrt系统固件	1.3.5
管理页面升级OpenWrt固件内核版本	1.3.6
怎样进入OpenWrt安全恢复模式	1.3.7
命令行 OpenWrt sysupgrade 升级固件版本	1.3.8
命令行uci设置OpenWrt Router模式拨号上网	1.3.9
命令行uci设置OpenWrt ap模式上网	1.3.10
OpenWrt 国内镜像源下载固件	1.3.11
shadowsocks-libev翻墙教程	1.4
什么是shadowsocks-libev翻墙软件	1.4.1
翻墙软件Shadowsocks-libev服务端设置	1.4.2
OpenWrt路由器运行shadowsocks-libev ss-local 客户端	1.4.3
史上最通俗易懂的OpenWrt翻墙路由器解释	1.4.4
配置OpenWrt路由器智能自动翻墙	1.4.5
OpenWrt自动更新设置和屏蔽广告	1.4.6
OpenWrt路由器翻墙为什么会失败	1.4.7
Shadowsocks不同加密算法的区别	1.4.8
零起点DO VPS shadowsocks-libev 翻墙设置教程	1.4.9
Android 安卓手机安装 shadowsocks 科学上网教程	1.4.10
OpenWrt + Git Bash for Windows 快速切换翻墙模式	1.4.11
编译、使用shadowsocks 翻墙软件	1.5
编译shadowsocks-libev for OpenWrt ipk安装包	1.5.1
下载和设置翻墙配置文件	1.5.2
使用Image Builder编译自动翻墙OpenWrt固件	1.5.3
如何使用预编译的OpenWrt翻墙固件	1.5.4
Ubuntu 使用 shadowsocks Simple-obfs obfs-server 混淆翻墙插件	1.5.5
深刻理解 shadowsocks simple-obfs 混淆插件工作原理	1.5.6
OpenWrt 路由器编译使用 Simple-obfs obfs-local 混淆插件	1.5.7
Windows PC翻墙最好方法:shadowsocks-libev + simple-obfs + TFO	1.5.8
应用: Netgear WNDR4300刷OpenWrt翻墙教程	1.6
WNDR4300 下载和设置Image Builder	1.6.1
WNDR4300 编译shadowsocks-libev ipk	1.6.2
WNDR4300 修改翻墙配置文件	1.6.3
WNDR4300 编译自动翻墙固件	1.6.4
WNDR4300 怎样刷自动翻墙固件	1.6.5
WNDR4300 登录并设置翻墙固件	1.6.6

应用:D-Link DIR-505刷OpenWrt翻墙教程	1.7
如何进入 DIR-505 恢复模式	1.7.1
DIR-505 刷OpenWrt固件过程	1.7.2
DIR-505 启用工作模式开关	1.7.3
DIR-505 Router 模式翻墙教程	1.7.4
DIR-505 AP 模式翻墙教程	1.7.5
DIR-505 编译OpenWrt全自动翻墙固件	1.7.6
DIR-505 刷预编译OpenWrt翻墙固件	1.7.7
登录并设置 DIR-505 OpenWrt 翻墙固件	1.7.8
其他翻墙软件使用教程	1.8
利用lantern 蓝灯实现浏览器自动翻墙	1.8.1
加强翻墙上网的匿名性	1.8.2
浏览器使用 DNS over HTTPS (DoH) 进行安全DNS	1.8.3
全面优化 Linux 系统	1.9
Ubuntu OpenWrt 开启 TCP Fast Open (TFO)流量加速	1.9.1
Shadowsocks 服务端 Ubuntu 开启BBR加速	1.9.2
Ubuntu server 最大打开文件数目优化	1.9.3
Linux TCP UDP 网络性能优化	1.9.4
Linux swap 交换文件优化	1.9.5
附录	1.10
翻墙软件、教程汇总	1.10.1
本机阅读本教程的方法	1.10.2
知识若不分享, 实在没有意义	1.10.3
如何贡献本项目	1.10.4

最好的 OpenWrt 路由器 shadowsocks 自动翻墙、科学上网教程

手把手教你路由器刷OpenWrt固件, 自动穿越万里长城

本科学上网方案的特点

放弃建立被墙网站黑名单的方案吧, 被墙的网站每天在增加, 黑名单永远无法完善

大道至简, 一劳永逸!

- 建立国内重要网站白名单, 在国内进行dns查询
- 其他网站通过通过 shadowsocks 服务端进行dns查询
- 亚洲或国内的IP流量走国内通道
- 其他流量通过shadowsocks服务端转发
- 路由器屏蔽国内外的广告
- 利用 Bash 一键切换翻墙模式

知识若不分享, 实在没有意义

2014年6月 Dropbox壮烈被墙

查资料发现, 著名的开源路由器固件OpenWrt支持家里的路由器 TP-Link WR2543N V1, 于是就给路由器安装了OpenWrt并设置为自动智能翻墙

再也没有打不开的网站了, 自由的感觉真好: youtube, twitter, facebook, google...

什么是圣人, 圣人就是得到和付出比较均衡的人:

- 天地生我, 我敬天地
- 父母育我, 我养父母
- 网上获得知识, 网上分享知识

于是, 花了许多天, 查资料, 写教程, 调试固件, 不知不觉一天就过去了

希望你应用本教程后, 也把你的过程写下来, 合并到这个项目中来:

<https://github.com/softwaredownload/openwrt-fanqiang>

Linux下如何编译OpenWrt shadowsocks自动翻墙固件

- 首先把本项目clone到本地目录, 如 ~/Downloads/openwrt-fanqiang
- 原始配置文件
 - ~/Downloads/openwrt-fanqiang/openwrt/default 默认配置文件夹
 - ~/Downloads/openwrt-fanqiang/openwrt/wndr4300 针对特定路由器型号的配置文件, 此处以wndr4300为例
- 复制配置文件
 - 本地建立配置文件目录, 如 ~/Downloads/openwrt-wndr4300
 - 复制默认配置文件夹下面的文件到 ~/Downloads/openwrt-wndr4300/ 下
 - 如果有针对特定路由器的配置文件, 也复制到~/Downloads/openwrt-wndr4300/, 并覆盖同名文件
- 修改配置文件, 编译后就直接可以用了。否则刷上固件后登录路由器再修改。主要修改:
 - openwrt-wndr4300/etc/shadowsocks-libev/config.json
 - openwrt-wndr4300/usr/bin/ss-firewall-asia
 - openwrt-wndr4300/etc/uci-defaults/defaults
- 编译自定义固件, 设置FILES=~/Downloads/openwrt-wndr4300

本项目规定的默认值

```
shadowsocks server:      1.0.9.8
shadowsocks server_port: 1098
shadowsocks local_port:  7654
shadowsocks tunnel_port: 3210
shadowsocks password:    killgfw
root login password:     fanqiang
```

WIFI password: icanfly9876

🔧 如何使用预编译翻墙固件：

- shadowsocks 服务端保持默认值(除了server IP)
- 路由器刷OpenWrt shadowsocks翻墙固件
- 登录路由器修改server IP：

```
# Modify 1.0.9.8 to your server IP address
vi /etc/shadowsocks-libev/config.json
# Modify 1.0.9.8 to your server IP address
vi /usr/bin/ss-firewall-asia
/etc/init.d/shadowsocks restart
```

- 以上修改测试通过后，建议再修改 shadowsocks password, 路由器root password
- 少数时候需要重启路由器才能使修改生效

🏠 关于 IPv6

默认翻墙固件不支持IPv6

有的软件如 Dropbox 桌面客户端默认连接到服务端 IPv6 地址，ping dropbox.com 出来的是IPv6 地址，可能导致客户端连接服务器失败，浏览器导航到 www.Dropbox.com 连接被重置

解决办法: 网络连接的属性，不要勾选 Internet Protocol Version 6 (TCP/IPv6)

📚 相关资源

- Netgear WNDR4300 预编译翻墙固件，支持xchacha20-ietf-poly1305(2018-10-22):
<https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/>
- shadowsocks-libev_3.2.0-1_mips_24kc.ipk, simple-obfs_0.0.5-3_mips_24kc.ipk (2018-10-22):
<https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>
- 史上最详细的OpenWrt路由器翻墙教程下载 PDF epub (2018-10)
<https://software-download.name/2014/fanqiang-jiaocheng/>
- Shadowsocks-libev Windows 客户端下载: ss-redir ss-tunnel obfs-local (2018-08 by cokebar)
<https://software-download.name/2018/shadowsocks-libev-windows-binary-download/>
- D-Link DIR-505 预编译翻墙固件 (2018-10-22):
<https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>
- TP-Link TLWR2543 预编译翻墙固件 (2018-10-22):
<https://software-download.name/2014/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade-bin-with-shadowsocks/>

😊 授权许可

除特别声明外，本书中的内容使用CC BY-SA 3.0 License(创作共用 署名-相同方式共享3.0许可协议)授权，代码遵循BSD 3-Clause License(3项条款的BSD许可协议)。







📖 在线阅读史上最详细的科学上网教程

- <https://fanqiang.software-download.name>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2019-07-03

最好的 OpenWrt 路由器 shadowsocks 自动翻墙、科学上网教程

- 🧑‍🔬 本科学上网方案的特点
- 😊 知识若不分享，实在没有意义
- 🐧 Linux下如何编译OpenWrt shadowsocks自动翻墙固件

-  本项目规定的默认值
-  如何使用预编译翻墙固件：
-  关于 IPv6
-  相关资源
-  授权许可
-  在线阅读史上最详细的科学上网教程

无线路由器刷OpenWrt固件的准备工作

在给您的路由器刷新固件之前，有必要先了解：

1. 什么是无线路由器固件
2. 准备支持OpenWrt路由器
3. 如何备份路由器配置

最简单的路由器刷OpenWrt翻墙方案：

- <https://github.com/softwaredownload/openwrt-fanqiang>

在线阅读OpenWrt路由器翻墙、科学上网教程：

- <https://fanqiang.software-download.name>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-10-22

什么是无线路由器固件

网络的本质是知识的开放与共享。人类社会进步速度, 如果原来是自行车速, 加上网络后, 就坐上了火箭

一个热爱学习的人, 必然要查找一些英文学习资料, 在某个国家的某个阶段必然会遇到的问题: 怎么Google搜索这么烂, 经常打不开, YouTube真差劲, 加载半天还在打转...

后来, 可能会发现, 不是人家烂, 而是有人故意为之

怎么办呢? 有很多种办法解决这个问题, 其中一个较好的方案是从家用无线路由器上解决, 然后全部有线和无线设备都可以无障碍上网了

路由器的原厂固件限制了用户自行开发功能, 我们必须给路由器刷上特定的固件, 并进行一些设置才可以翻墙

无线路由器就好比是一台小电脑。电脑上安装了Windows XP, Windows 7, Windows 8, 或者Ubuntu等操作系统就可以使用了。固件就是给路由器使用的操作系统, 是固化在路由器芯片内的操作系统

常用的开源第三方无线路由器固件

- 开源OpenWRT路由器固件: 部署复杂、灵活性高

这也是本文系列所用的固件。发展成熟, 支持的硬件多

- 开源DD-WRT路由器固件: 支持广泛、功能全面

DD-WRT比较实用, 通过网页对固件进行配置的功能强大, 但是定制和扩展比较困难

- 开源Tomato路由器固件: 衍生版本众多

原始版本固件代码自2010年后就再也没有更新

本系列教程使用OpenWrt来讲解路由器翻墙方法

相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

[什么是无线路由器固件](#)

-  常用的开源第三方无线路由器固件

支持刷OpenWrt路由器列表推荐

现在移动设备已经普及，一般情况下读者家里都已经有无线路由器了，到底能不能刷上OpenWrt固件呢？到OpenWrt官方网站查一下就知道了

打开 [支持OpenWrt无线路由器列表](#) 这个页面，搜索一下。比如我家原来的无线路由器型号是 TP-LINK TL-WR2543N，同时按下Ctrl+F，输入 **WR2543** 就可以找到,如下图：

TL-WR2543ND	1.0	12.09	ar71xx	Atheros AR7242	400	8	64	(V
Model	Version	Status	Target(s)	Platform	CPU Speed (MHz)	Flash (MB)	RAM (MB)	V

从上图可以看出，OpenWrt支持 WR2543N 无线路由器版本1。此外，还可以看出更多信息，比如芯片类型是ar71xx，芯片型号是Atheros AR7242，CPU频率是400 MHz，原厂带8MB Flash，64MB RAM内存

目前 WR2543N已经比较少见。如果你购买其他品牌，建议Flash在8 MB或以上，RAM在64MB以上

如果你准备买新路由器，可以在上面列表中查找OpenWrt推荐路由器型号，能买到的话，再以关键词 型号 **OpenWrt** 在搜索引擎搜索相关信息，确保你想购买的型号能比较容易地刷上 OpenWrt固件

作为新手来说，推荐使用 D-Link DIR-505，可能是最便宜的适合学习OpenWrt的路由器

相关资源：

- <https://fanqiang.software-download.name/>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-12-07

怎样备份原厂路由器配置文件

提示, 刷机有风险, 如果不当操作, 或者有其他意外发生, 路由器可能变成砖头, 本文系列旨在提供参考, 刷机风险由读者自负, 作者不承担任何责任, 也没有义务提供个别指导

本文作者给 WR2543N 刷 OpenWrt 固件不下10次, 因为完全没有经验, 有几次刷了后不能进入管理界面, 只能用手机3G上网查找解决方案, 还好 WR2543N 非常容易进入安全模式, 然后重新刷固件, 解决了问题。作为初学者, 一定要购买容易进入安全模式的路由器

对于本文作者来说, 现在已经不需要原厂固件了, 但是在第一次刷OpenWrt前, 我还是把原厂固件的配置文件作备份, 建议读者也是如此

怎样备份原厂固件, WR2543N的原厂说明书说得 very 详细, 建议找出来详细阅读

LAN 和WAN的区别

什么是LAN和WAN,第一次听到这种专业名词容易让人头大

LAN并不是一个单词, 而是三个英文单词的缩写: Local Area Network, 查出这三个单词的意思, 就比较好理解了, 就是 本地区域网络 的意思。本地, 比如是室内, 公司内, 办公室内都是本地, 也就是LAN是用来连接本地电脑的

WAN, Wide Area Network, 广泛区域网络, 也就是连向更广泛的外部的网络, 一般家用就是通向ADSL modem, 再通过ADSL modem连接互联网

路由器通常有多个LAN口, 一个WAN口

在WR2543N路由器的后背, 有并排4个的网线插口, 叫LAN口, 单独的一个网线插口叫WAN口, WAN口旁边还有个USB插口。把ADSL的线插在WAN口。备好一根网线, 一头插路由器的任意一个LAN口, 另一头插电脑

设置电脑LAN口IP地址

路由器和电脑都处在本地网络里面, 为了互相区分, 本地网络的每台设备都需要有不同的IP地址

本路由器默认 LAN 口 IP 地址是 192.168.1.1, 默认子网掩码是 255.255.255.0

电脑的IP地址要和路由器的不同, 我们可以设置电脑的本地IP地址为动态获取。如果手动设置IP地址, 那么计算机IP地址必须为192.168.1.X (X) 是2到254之间的任意整数), 子网掩码须设置为255.255.255.0, 默认网关须设置为192.168.1.1

以Windows XP 系统为例, 介绍计算机参数的设置步骤

右键单击桌面上的 网上邻居 图标, 选择 属性, 在打开的 网络连接页面中, 右键单击“本地连接”, 选择状态, 打开“本地连接状态”进行操作。详细步骤请见购机时附带的手册

登录路由器管理界面

打开网页浏览器，在浏览器的地址栏中输入路由器的 IP 地址：192.168.1.1，可以看到下图：



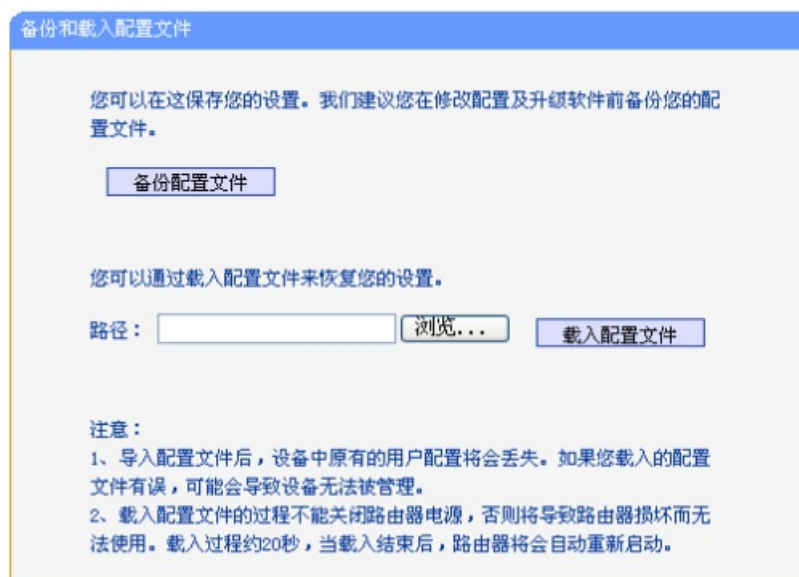
所示登录界面，输入用户名和密码（用户名和密码的出厂默认值均为admin），单击确定按钮

备份原厂路由器固件配置文件

登录路由器管理界面后，选择菜单，系统工具→备份和载入配置，可以在如下图所示备份或载入路由器配置文件

配置备份功能可以将路由器的设置以文件形式保存到电脑中，以备下次使用；在升级路由器软件或在载入新的配置文件前备份路由器的原有配置，可以有效防止升级软件或载入新配置文件过程中丢失原有配置的问题

配置载入功能则可以将先前保存的或已编辑好的配置文件重新载入







相关资源：

- <https://fanqiang.software-download.name/>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-12-07

怎样备份原厂路由器配置文件

-  LAN 和WAN的区别
-  设置电脑LAN口IP地址
-  登录路由器管理界面
-  备份原厂路由器固件配置文件

路由器怎么刷 OpenWrt 固件教程

经过前面的准备, 终于要给亲自给路由器刷OpenWrt固件了。有可能失败, 有可能成功。一连嘴里念叨FGW (=fuck great wall),一边给自己打气

OpenWrt有必要装中文管理界面吗

我认为不需要。网上最新最全面的信息都是英文的。GFW在不断进步, 我们也要不停地学习。我们要感谢GFW, 让我们每天多记几个单词。一些步骤的操作, 我特意截图并加上了步骤标识, 实在记不住就每次打开这个教程照着图示来

在开源的Linux类操作系统里连接OpenWrt进行操作

我认为有必要从现在开始切换到Linux类操作系统了。Windows已经开始走向没落, 开源操作系统渐渐赶上闭源商业操作系统

为什么呢? 随着技术的不断进化, 开源的技术合作越来越方便。打个比方, 如果佛教老大释迦牟尼, 基督教创始人耶稣在世, 不开源恐怕也会穷途末路

再说OpenWrt就是微型的Linux操作系统, 熟悉了Linux, 学习OpenWrt就很容易了

在以后的教程里, 都是在Ubuntu下对OpenWrt进行管理。如果有两台电脑, 建议一台装Ubuntu, 如果只有一台电脑, 可以装Ubuntu和Windows双启动

最简单的路由器刷OpenWrt翻墙方案:

- <https://github.com/softwaredownload/openwrt-fanqiang>

在线阅读OpenWrt路由器翻墙、科学上网教程:

- <https://fanqiang.software-download.name>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07
路由器怎么刷 OpenWrt 固件教程

-  OpenWrt有必要装中文管理界面吗
-  在开源的Linux类操作系统里连接OpenWrt进行操作

怎样从官网下载OpenWrt固件

☺ 从官网下载最新版的适合自己路由器的OpenWRT固件

- 进入OpenWrt固件下载主页面:

<http://downloads.openwrt.org/>

截止2018-09, 最新稳定发行版:

```
OpenWrt 18.06.1
Released: Sat, 18 Aug 2018
```

Development Snapshots是开发版, 包含最新的功能, 但可能不够稳定

<http://downloads.openwrt.org/snapshots/targets/>

如果使用Snapshots没有什么问题, 当然是最好的选择, 否则可以尝试一下稳定发行版

下面以稳定版和WR2543举例

- 选择路由器的CPU类型

打开页面后, 选择你的路由器的芯片型号进入, 很多是ar71xx系列, 于是进入了:

<http://downloads.openwrt.org/snapshots/targets/ar71xx/>

- 选择路由器的Flash类型

再选择Flash类型, 比如WR2543是generic, 网件WNDR4300路由器是nand

<http://downloads.openwrt.org/snapshots/targets/ar71xx/generic/>

再选择你的路由器型号, 页面搜索 wr2543, 找到了吗。有两个文字供下载, 一个文件结尾是 factory.bin, 适合原厂固件下刷, 另一个文件名结尾是 sysupgrade.bin, 适合已经是OpenWrt系统下刷

☺ OpenWrt官方wiki下载OpenWrt固件 for WR2543

OpenWrt官方网页上有WR2543N的专页, 详细介绍了刷机步骤及注意事项。

打开官方Wiki页面 [TP-Link TL-WR2543ND](#)

上面列出了支持的版本: v1.0和v1.2。我的路由器是v1.0的, 可以刷, 你的版本如果不是这两个, 不能确保能刷成功

这两个固件都带LuCI 网页管理界面。有时候, 如果你升级了不带LuCI的固件, 命令行方式又无法搞定OpenWRT上网参数设置, 就需要先在电脑里下载带LuCI的固件, scp复制到路由器升级, 再通过网页设置

有两个固件供下载:

- [openwrt-ar71xx-generic-tl-wr2543n-v1-squashfs-factory.bin](#) - Installing OpenWRT from factory
- [openwrt-ar71xx-generic-tl-wr2543n-v1-squashfs-sysupgrade.bin](#) - Upgrading an existing OpenWRT install

一定要注意:

- 在原厂固件上刷OpenWrt, 要用固件文件名带 **factory** 的.bin文件
- 已经刷了OpenWrt固件, 再升级 OpenWrt固件时就要用文件名带 **sysupgrade** 的 .bin文件

现在我们是在原厂固件基础上刷 OpenWrt, 自然是下载第一个文件, 也就是 openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-factory.bin

要确保下载下来的文件完整, 下载过程没有中断, 如果下载下来的文件不完整, 并把这个不完整的文件刷进机器, 恢复起来很麻烦, 有可能变砖

- Netgear WNDR4300 预编译翻墙固件, 支持xchacha20-ietf-poly1305(2018-10-22):
<https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/>
- D-Link DIR-505 预编译翻墙固件 (2018-10-22):
<https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>
- TP-Link TLWR2543 预编译翻墙固件 (2018-10-22):
<https://software-download.name/2014/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade-bin-with-shadowsocks/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name>
怎样从官网下载OpenWrt固件

2018-12-07

-  从官网下载最新版的适合自己路由器的OpenWRT固件
-  OpenWrt官方wiki下载OpenWrt固件 for WR2543

进管理页面刷OpenWrt教程:WR2543路由器为例

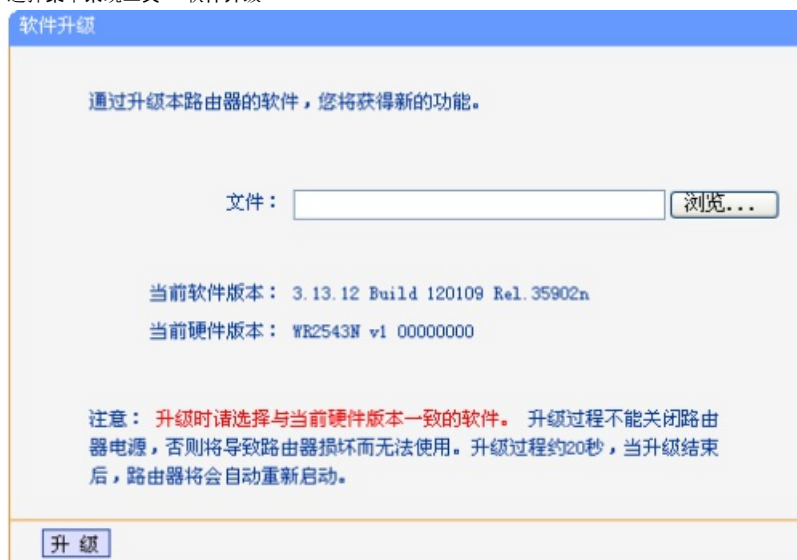
通过有线或无线登录WR2543路由器管理页面

打开浏览器,输入路由器的IP地址: 192.168.1.1

回车,在密码验证框,输入用户名: admin 密码也是 admin

进路由器管理页面进行器固件升级

选择菜单系统工具→ 软件升级



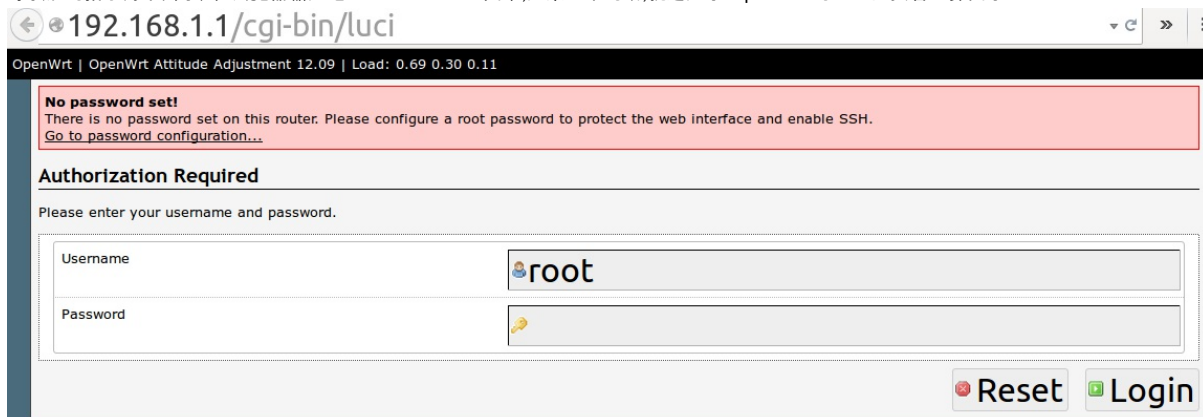
点击 浏览 按钮选择下载的文件 openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-factory.bin

注意, 文件名必须是...factory.bin

再单击 升级 进行软件升级。要注意, 在刷固件过程中不可停电或其他原因造成中断, 否则路由器就变砖了

等待几分钟

等锁形的指示灯不闪了, 在浏览器输入地址: 192.168.1.1 回车,如果正常的话,就进入了 OpenWrt 的LuCI网页管理界面了



默认用户名是root,默认密码是空。点 Login 直接登录

相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name>

2018-12-07

进管理页面刷OpenWrt教程:WR2543路由器为例

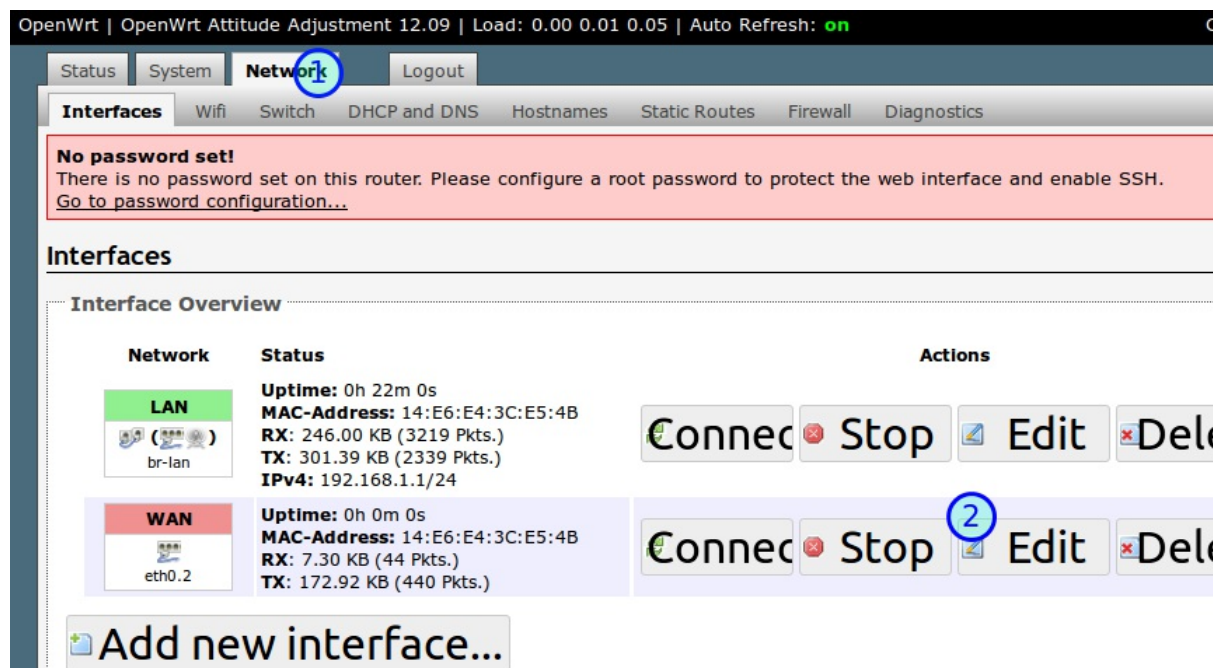
-  通过有线或无线登录WR2543路由器管理页面
-  打开浏览器,输入路由器的IP地址: 192.168.1.1
-  进路由器管理页面进行器固件升级

管理页面OpenWrt PPPOE自动拨号上网设置教程

见面界面登录路由器后, 就可以设置上网参数了

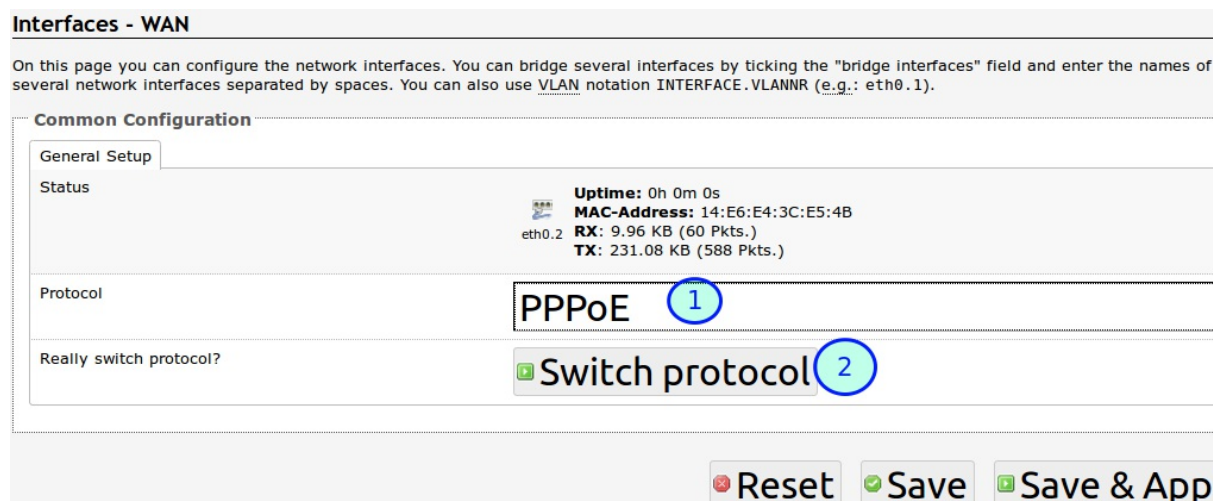
管理后台编辑OpenWrt WAN上网设置

选择上面的 Network, 在 Interface里, WAN右边, 选择Edit。WAN和ADSL modem相连, 设置拨号上网自然是在WAN而不是LAN。



管理页面配置OpenWrt PPPOE 自动拨号上网

进去后, 在协议 Protocol 下拉列表框里, 选择拨号上网的协议, 也就是 PPPoE, 再点击下面的 Switch Protocol切换协议



管理后台设置 OpenWrt PPPOE 自动拨号上网用户名、密码

1. PAP/CHAP username: 拨号上网用户名
2. PAP/CHAP password: 拨号上网密码
3. 点击 **Save & Apply** 保存并应用设置

Common Configuration

General Setup

Advanced Settings

Physical Settings

Firewall Settings

Status

pppoe-wan

RX: 0.00 B (0 Pkts.)
TX: 0.00 B (0 Pkts.)

Protocol

PPPoE

PAP/CHAP username

admin 1

PAP/CHAP password

••••••••

2

Access Concentrator

auto

Leave empty to autodetect

Service Name

auto

Leave empty to autodetect

Reset

Save

3 Save & Apply

这时，连接LAN的电脑应该已经可以上网了，但无线设备还不行

相关资源：

- <https://fanqiang.software-download.name/>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-12-07

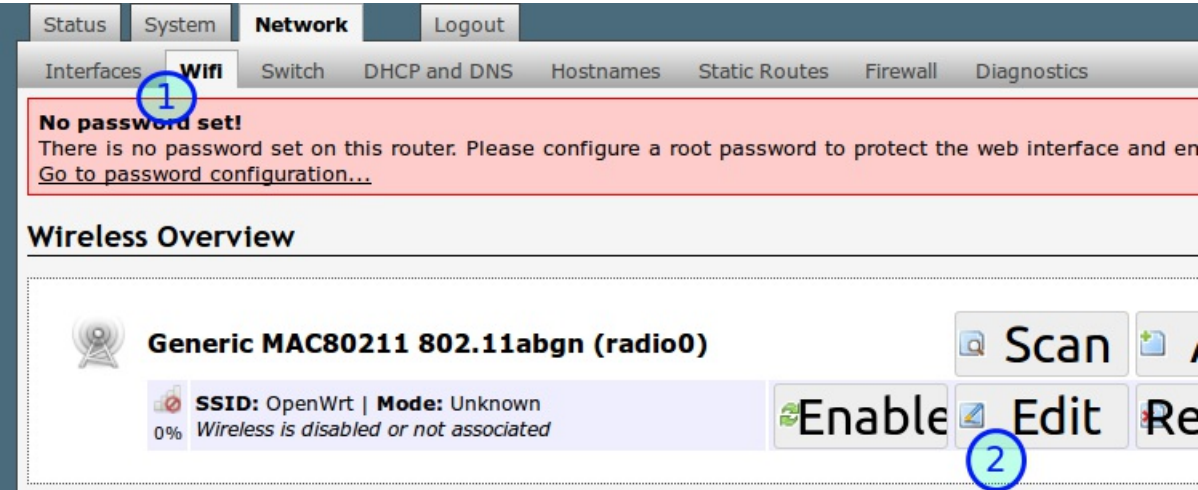
[管理页面OpenWrt PPPOE自动拨号上网设置教程](#)

- [管理后台编辑OpenWrt WAN上网设置](#)
- [管理页面配置OpenWrt PPPOE 自动拨号上网](#)
- [管理后台设置 OpenWrt PPPOE 自动拨号上网用户名、密码](#)

管理页面OpenWrt开启、设置无线(Wifi)教程

登录OpenWrt路由器管理后台后：

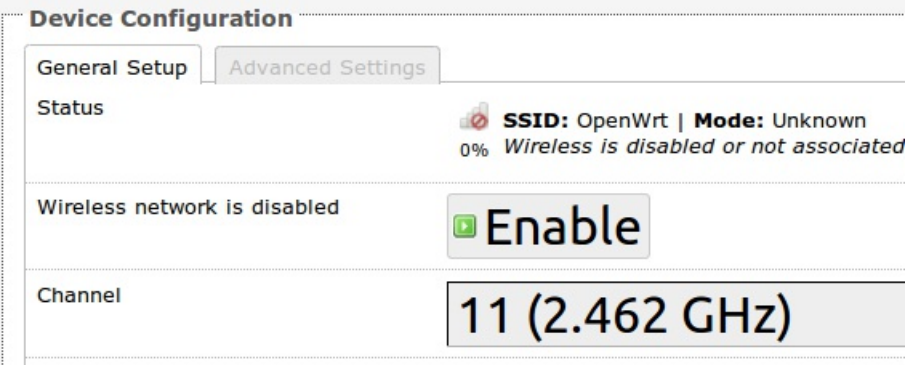
 选择 **Network, Wifi, Edit**



点击Enable按钮，这时无线设备已经可以连上Wifi

Wireless Network: Unknown "OpenWrt" (radio0.network1)

The *Device Configuration* section covers physical settings of the radio hardware such channel, transmit power or antenna selection which is shared among all defined wireless networks (if the radio hardware is multi-SSID capable). Per network settings like encryption mode are grouped in the *Interface Configuration*.



默认ESSID就是OpenWrt，没有密码。不想做活雷锋的加个密码吧

OpenWrt Wifi密码设置

把ESSID改成 eastking-wr2543,然后：

- 点击Wireless Security进入OpenWrt无线安全设置
- Encryption加密方式，WPA2-PSK
- Key密码:killgw
- Save & Apply 保存并应用设置

Transmit Power 27 dBm (501 mW)

Interface Configuration

General Setup **Wireless Security** MAC-Filter

Encryption WPA2-PSK

Cipher auto

Key •••••

Reset Save Save & Apply

这时，所有无线设备都可以通过OpenWrt路由器上网了

🔑 OpenWrt管理界面登录密码设置

你注意到没有，网页上方有一个红色的框框(No password set!)一直在提示我们：小人不得不防，OpenWrt叫你设一个路由器管理界面登录密码呢！

1. 点击最上面的System进入系统设置
2. 再点击Administration进入管理员设置
3. 密码Password: fanqiang
4. 确认密码Confirmation: fanqiang

OpenWrt | OpenWrt Attitude Adjustment 12.09 | Load: 0.00 0.01 0.06

Status **System** Network Logout

System **Administration** Software Startup Scheduled Tasks LED Configuration

No password set!
There is no password set on this router. Please configure a root password to protect the web interface.
[Go to password configuration...](#)

Router Password

Changes the administrator password for accessing the device

Password •••••

Confirmation •••••

5. 其他设置：下面的：

Gateway ports, 勾选 **Allow remote hosts to connect to local SSH forwarded ports** (允许远程主机连接本地SSH转发端口), 这样我们就可以用SSH命令行的方式管理路由器。最后点击右下角 Save & Apply保存并应用设置

相关资源：

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name>
管理页面OpenWrt开启、设置无线(Wifi)教程

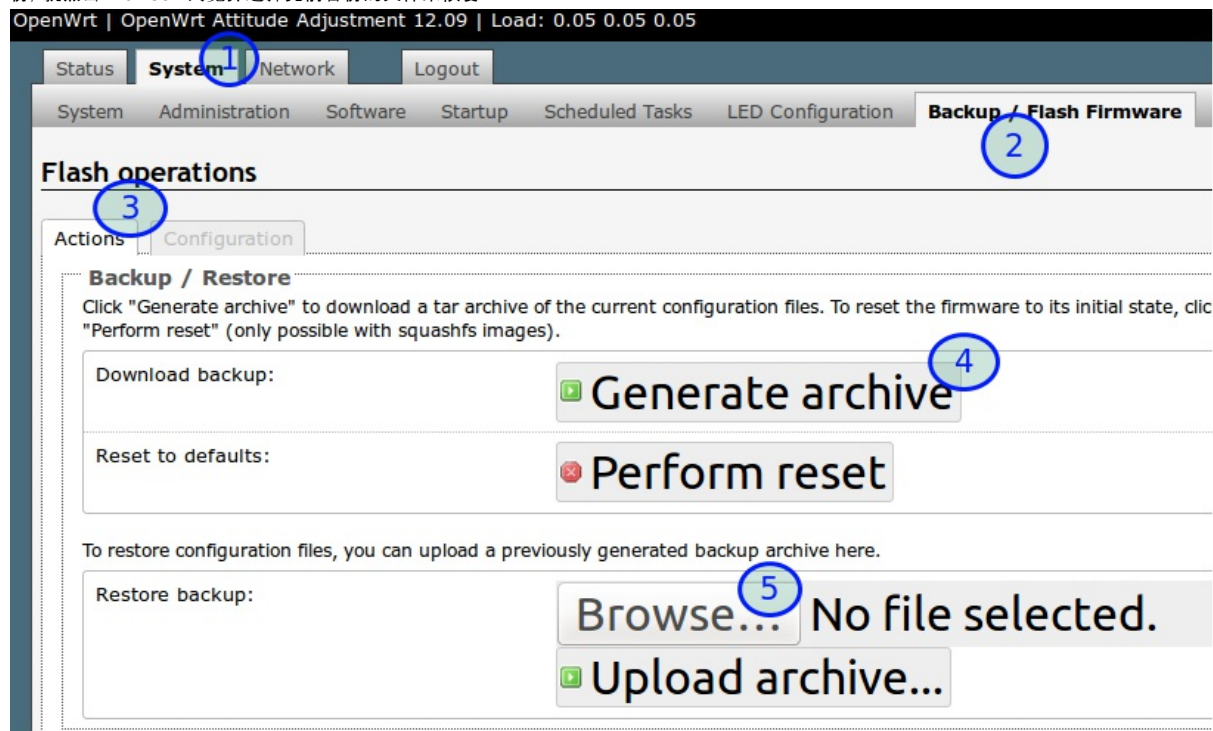
2018-12-07

-  选择 Network, Wifi, Edit
-  OpenWrt Wifi密码设置
-  OpenWrt管理界面登录密码设置

管理页面备份OpenWrt系统固件

现在有线和无线上网都正常了。应该把现有的OpenWrt设置备份一下, 因为我们还要经常折腾OpenWrt, 有时一个设置错误, 可能就上不了网, 有了备份, 就可以快速恢复

选择**System**系统设置 选择**Backup / Flash Firmware**备份恢复固件 **Actions**动作 **Generate**生成备份文件并保存到电脑 如果以后你要恢复备份, 就点击**Browse...**浏览并选择先前备份的文件来恢复



相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name>

2018-12-07

管理页面LuCI升级OpenWrt固件内核版本

我们现在已经给TP-Link WR2543N刷上了OpenWrt固件，并且可以正常上网了。如果要升级OpenWrt固件，又该怎么做呢？

有两个途径升级固件：

- LuCI web界面升级
- SSH命令行登录路由器升级

本节就讲 web管理界面LuCI升级固件的方法

📁 下载OpenWrt升级用固件sysupgrade.bin

下载用于WR2543N路由器的升级固件，升级用固件文件名中有sysupgrade字样

还是到OpenWrt Wiki页面 [TP-Link TL-WR2543ND](#)

点击下载链接，比如 <http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/generic/openwrt-18.06.1-ar71xx-generic-tl-wr2543-v1-squashfs-factory.bin> 其实这个固件的核心和我们先前安装的...factory.bin一样，我们是出于实验目的，演示升级固件的方法。

🌐 用前文讲过的方法从网页登录OpenWrt路由器

👤 LuCI 开始升级OpenWrt固件内核版本

1. System系统
2. Backup / Flash Firmware备份或刷新固件
3. Flash new firmware, Browse...选择我们刚下载下来的固件
4. Flash image...刷新固件

注：如果Keep settings保持勾选，升级固件后，原来的设置就会保留，不用重新设置拨号上网参数




The screenshot shows the OpenWrt LuCI web interface. The top navigation bar includes 'Status', 'System', 'Network', and 'Logout'. The 'System' tab is selected, and the 'Backup / Flash Firmware' sub-tab is active. The 'Flash operations' section is visible, containing 'Backup / Restore' and 'Flash new firmware image' sections. The 'Flash new firmware image' section has a 'Keep settings' checkbox checked and a 'Flash image...' button. The 'Flash image...' button is circled in blue and labeled with a blue '4'. The 'Browse...' button next to it is circled in blue and labeled with a blue '3'. The 'Flash image...' button is also labeled with a blue '4'.

相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

[管理页面](#)[LuCI升级OpenWrt固件内核版本](#)

-  下载OpenWrt升级用固件sysupgrade.bin
-  用前文讲过的方法从网页登录OpenWrt路由器
-  LuCI 开始升级OpenWrt固件内核版本

怎样进入OpenWrt 安全恢复模式(WR2543N为例)

有时候, 我们可能操作失误, 无法进入LuCI网页界面管理恢复固件, 这时就需要进入安全模式来恢复了

不同的路由器, 进入安全模式的方法可能有所差别, 本文系列适用于 TP-LINK WR2543N

安全模式是玩OpenWrt的救命仙丹。能熟练进入安全模式来恢复设置, 是OpenWrt已经上手的一个标志

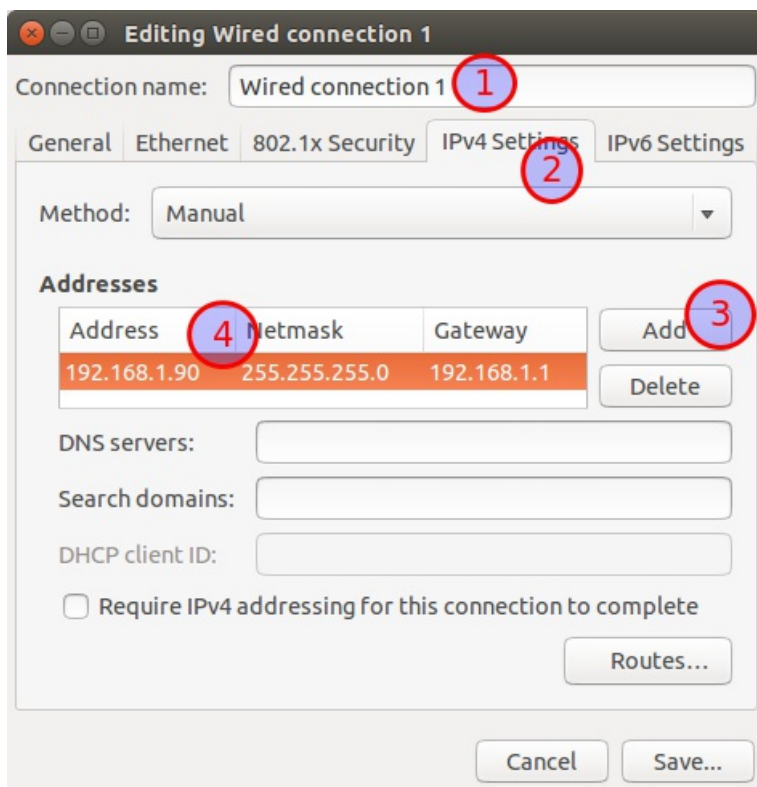
进入安全模式时, 没有无线连接可用, 所以我们要有线的方式登录OpenWrt。OpenWrt默认的IP地址是192.168.1.1, 我们要设置电脑有线连接的IP地址类似于192.168.1.x, 其中x是2至255的数字

WR2543N无线路由器进入OpenWrt安全模式的方法:

1. 用网线把路由器和电脑连接起来, 设置电脑网卡的IPv4地址

以Ubuntu为例, 点击桌面右上角连接符号, 选择 **Edit Connections**, 再选择 Ethernet连接, 点击 Edit 按钮, 在弹出的窗口中选择 IPv4 Settings, Method选择Manual, Address栏点击Add, 设置如下:

- Address: 192.168.1.97
- Netmask: 255.255.255.0
- Gateway: 192.168.1.1



2. 在Ubuntu运行命令:

```
sudo tcpdump -Ani eth0 port 4919 and udp
```

3. 重启路由器, 当WR2543N的锁形指示灯刚开始闪烁时, 立即按路由器背面的wps按钮3次

4. Ubuntu命令行界面出现:

```
Please press button now to enter failsafe
```

```
donald@Software-Download:/ $ sudo tcpdump -Ani eth0 port 4919 and udp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
20:08:41.122133 IP 192.168.1.1.46561 > 192.168.1.255.4919: UDP, length 1001
E.....@.@.....7..!...Please press button now to enter failsafe.....
```

5. Ubuntu命令行执行(有时可以不需tcpdump直接telnet):

```
telnet 192.168.1.1
```

这时就成功登录了OpenWrt, 如下图:

```
dona1d@Software-Download:/# telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.

=== IMPORTANT =====
Use 'passwd' to set your login password
this will disable telnet and enable SSH
-----

github.com/softwaredownload/openwrt-fanqiang

BusyBox v1.19.4 (2013-03-14 11:28:31 UTC) built-in shell (ash)
Enter 'help' for a list of built-in commands.


|_| .------. | |_| | | .----. | |_| | | | | | | | | | |
| - || _ | -_|| | | | | _|| |
|_| || _|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
      |__| W I R E L E S S   F R E E D O M

-----
ATTITUDE ADJUSTMENT (12.09, r36088)
-----
* 1/4 oz Vodka          Pour all ingredients into mixing
* 1/4 oz Gin            tin with ice, strain into glass.
* 1/4 oz Amaretto
* 1/4 oz Triple sec
* 1/4 oz Peach schnapps
* 1/4 oz Sour mix
* 1 splash Cranberry juice
-----

root@(none):/# █
```

6. 设置登录OpenWrt SSH登录密码:

```
passwd
#输入密码 fanqiang
```

如果出现：

```
passwd: /etc/passwd: Read-only file system
passwd: can't update passwd file /etc/passwd
```

就输入 `mount_root` 再重新passwd设置管理员密码

如下图:

```
-----
root@(none):/# passwd
Changing password for root
New password:
Retype password:
passwd: /etc/passwd: Read-only file system
passwd: can't update password file /etc/passwd
root@(none):/# mount_root
switching to jffs2
root@(none):/# passwd
Changing password for root
New password:
Retype password:
Password for root changed by root
root@(none):/# █
```

telnet登录路由器后, 可以用vi命令修改设置

这时如果你试图用浏览器登录192.168.1.1进入管理界面的话, 可能失败

重启路由器, 路由器锁形指示灯先是慢闪, 到变成常亮时, 你可以登录 192.168.1.1管理界面。一切恢复正常

相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name>

2018-12-07

OpenWrt sysupgrade 命令行升级固件内核版本

下面我们要使用 sysupgrade 更新固件到新版

要注意的是, 如果刷的是开发版, 可能不稳定, 刷机风险自己承担

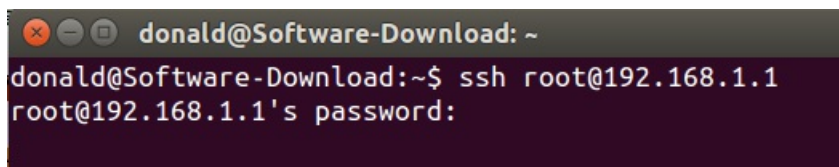
在浏览器里登录 192.168.1.1 进行固件升级是比较简单的。今天我们要尝试的是命令行刷机升级。命令行的方式更强大

SSH登录路由器

在Ubuntu里, 按Ctrl+Alt+T打开命令行终端, 输入:

```
ssh root@192.168.1.1
```

输入密码, 登录成功



```
donald@Software-Download: ~  
donald@Software-Download:~$ ssh root@192.168.1.1  
root@192.168.1.1's password:
```

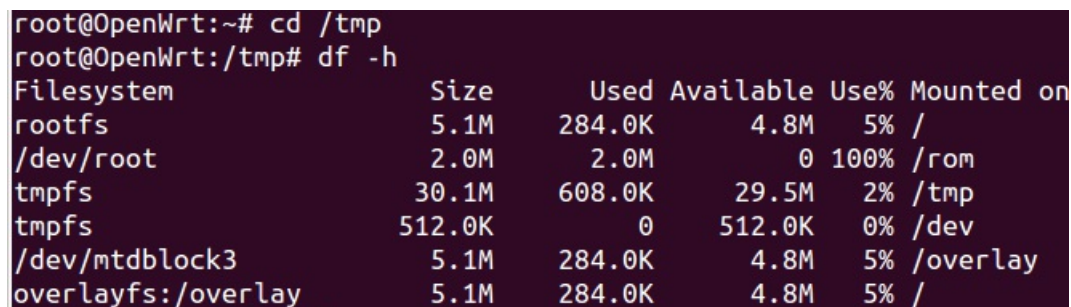
进入OpenWrt /tmp目录

```
cd /tmp
```

检查OpenWrt路由器是否有足够的内存

```
df -h
```

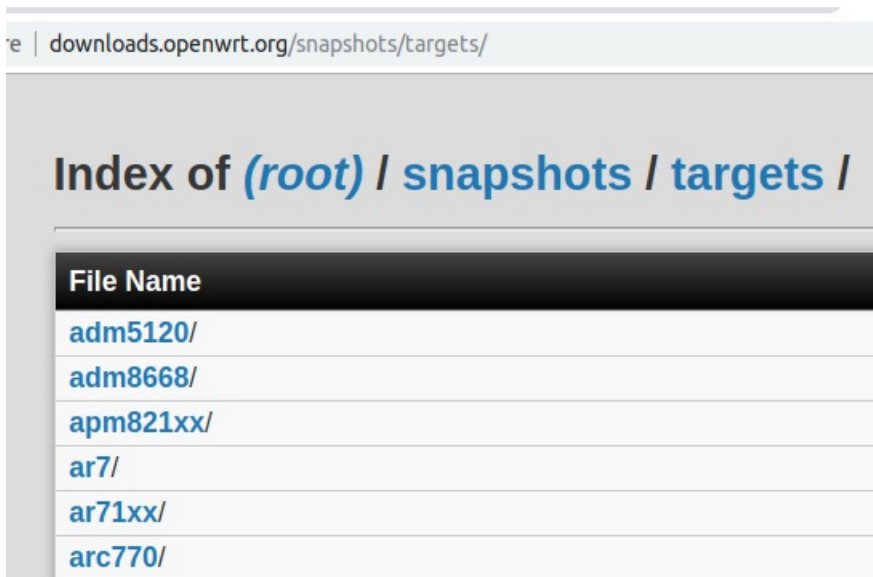
可以看出, /tmp 还有29.5MB可用空间, 而升级固件在3MB左右, 足够了



```
root@OpenWrt:~# cd /tmp  
root@OpenWrt:/tmp# df -h  
Filesystem      Size      Used Available Use% Mounted on  
rootfs          5.1M    284.0K      4.8M   5% /  
/dev/root       2.0M      2.0M        0 100% /rom  
tmpfs           30.1M    608.0K     29.5M   2% /tmp  
tmpfs           512.0K        0     512.0K   0% /dev  
/dev/mtdblock3  5.1M    284.0K      4.8M   5% /overlay  
overlayfs:/overlay 5.1M    284.0K      4.8M   5% /
```

下载OpenWrt最新trunk版本固件

- 在Ubuntu里浏览器打开 <http://downloads.openwrt.org/snapshots/targets/>
- TP-LINK WR2543N路由器的芯片类型是ar71xx, 就点击 [ar71xx](#) 目录进入。要注意, 路由器的芯片类型千万不能搞错, 不同路由器很可能是不同的



- TP-LINK WR2543路由器的Flash类型为 generic, 于是进入了 <http://downloads.openwrt.org/snapshots/targets/ar71xx/generic/>
- 按Ctrl+F查找自己的路由器型号。比如我输入的是 **wr2543**, 有两个固件, 升级用的是 **sysupgrade.bin**文件。右键点击该链接, 复制下载地址。在FireFox里是 **Copy Link Location**复制链接地址
- 回到Ubuntu命令行终端, 下载固件到 **/tmp** 目录。TP-LINK wr2543路由器是这样的:

```
root@OpenWrt:/tmp# wget http://downloads.openwrt.org/snapshots/targets/ar71xx/generic/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin
```

sha256校验, 确保下载的固件完整

```
root@OpenWrt:/tmp# wget http://downloads.openwrt.org/snapshots/targets/ar71xx/generic/sha256sums
root@OpenWrt:/tmp# sha256sum -c sha256sums 2> /dev/null | grep OK
openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin: OK
```

输出结尾是OK, 说明固件是完整的

OpenWrt sysupgrade命令升级OpenWrt固件

```
root@OpenWrt:/tmp# sysupgrade -v openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin
...
Upgrade completed
Rebooting system...
```




过约2分钟, 等路由器重启成功, 如果没有意外, 会发现有线和无线上网都正常。但浏览器192.168.1.1无法登录, 因为snapshots版本固件是不带LuCI网页管理界面的。没有也好, 可以节省路由器的存储空间, 也可以学习一下命令行管理OpenWrt路由器

相关资源:

- <https://openwrt.org/docs/guide-user/installation/generic.sysupgrade>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

OpenWrt sysupgrade 命令行升级固件内核版本

-  SSH登录路由器
-  进入OpenWrt /tmp目录
-  检查OpenWrt路由器是否有足够的内存
-  下载OpenWrt最新trunk版本固件
-  sha256校验, 确保下载的固件完整
-  OpenWrt sysupgrade命令升级OpenWrt固件

命令行 uci设置 OpenWrt router 模式拨号上网

如果路由器可以正常上网的前提, 我们可以ssh登录路由器, 直接在路由器的/tmp目录wget下载最新版固件并sysupgrade命令进行固件升级

有时候, 路由器无法上网, 这时候, 可以在电脑里下载好固件, 再把固件复制到路由器, 再sysupgrade升级或设置其他参数

只要能进入路由器的安全模式, 并ssh登录路由器, 一切都不是问题

Ubuntu下载OpenWrt for TP-LINK wr2543N trunk版固件

```
cd ~/Downloads
wget http://downloads.openwrt.org/snapshots/targets/ar71xx/generic/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin
```

scp复制固件到OpenWrt路由器 /tmp目录

```
scp openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin root@192.168.1.1:/tmp/
```

ssh登录OpenWrt路由器

```
ssh root@192.168.1.1
cd /tmp
```

sysupgrade升级固件并取消保留原来配置文件

注意, 升级后将无法上网, 也没有LuCI网页界面可以设置, 必须以命令行方式设置好上网参数

如果在下面的实验中, 命令行方式无法搞定路由器上网, 就只能在电脑里下载好带luCI的固件, scp复制固件到路由器升级固件, 然后以网页方式设置上网

在进行这一步前, 确保你熟练掌握以前部分教程

```
root@OpenWrt:/tmp# sysupgrade -n openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin
```

参数 `-n` 表示升级时不保留原来的配置文件。固件刷好后会自动重启, 这时要用前文教程讲过的方法进入OpenWrt安全模式, 登录路由器并重新设置root密码

下面假设你已经登录了路由器并设好了root密码

OpenWrt uci命令行设置拨号上网:

```
root@OpenWrt: uci set network.wan.proto='pppoe'
root@OpenWrt: uci set network.wan.username='wan-username'
root@OpenWrt: uci set network.wan.password='wan-password'
root@OpenWrt: uci set network.wan.peerdns=0
```

wan-username替换成你自己的拨号上网用户名, wan-password替换成你自己的密码

以上命令行的操作对象是文件 `/etc/config/network`

OpenWrt uci命令行设置无线上网:

```
root@OpenWrt: uci set wireless.@wifi-device[0].channel=11
root@OpenWrt: uci set wireless.@wifi-device[0].txpower=17
root@OpenWrt: uci set wireless.@wifi-device[0].disabled=0
root@OpenWrt: uci set wireless.@wifi-device[0].country='CN'
root@OpenWrt: uci set wireless.@wifi-iface[0].mode='ap'
root@OpenWrt: uci set wireless.@wifi-iface[0].ssid='eastking-tlwr2543'
```



```
root@OpenWrt: uci set wireless.@wifi-iface[0].encryption='psk2'
root@OpenWrt: uci set wireless.@wifi-iface[0].key='icanfly9876'
root@OpenWrt: uci commit wireless >/dev/null
```

以上命令实际上是应用在 /etc/config/wireless 文件上

uci设置说明:

- channel 信道
- txpower 功率
- disabled 是否启用无线, 0表示启用
- ssid 名称, 推荐后面以路由器型号结尾, 这样调试多个路由器时不会混淆
- encryption 加密方式
- key 无线密码, 如果你照上文的设置不动, 好处是忘记密码时可以上 <http://www.github.com/softwaredownload/openwrt-fanqiang> 来查看

允许远程主机用ssh的方式登录路由器及设置时区

```
root@OpenWrt: uci set dropbear.@dropbear[0].GatewayPorts='on'
root@OpenWrt: uci set system.@system[0].zonename='Asia/Shanghai'
root@OpenWrt: uci set system.@system[0].timezone='CST-8'
root@OpenWrt: uci commit system
```

ssh登录OpenWrt相关高级设置(你可能暂时用不到)

```
root@OpenWrt: uci set dropbear.@dropbear[0].Port=22
root@OpenWrt: uci set dropbear.@dropbear[0].PasswordAuth=off
root@OpenWrt: uci set dropbear.@dropbear[0].RootPasswordAuth=off
root@OpenWrt: uci commit dropbear
```

说明(不懂千万别乱设):

- Port ssh默认端口就是22,可以改成其他的提高安全性
- PasswordAuth ssh是否启用密码登录。如果你改成off, 又没有设置好ssh私钥和安装好LuCI, 你将无法ssh方式登录路由器, 唯一的办法就是安全恢复模式登录重新开始设置
- RootPasswordAuth 是否允许root用密码登录, 如果已经设置好了ssh私钥就可以改成off增加安全性

启用新的网络和无线设置

```
root@OpenWrt: /etc/init.d/dropbear restart
root@OpenWrt: /etc/init.d/system restart
root@OpenWrt: /etc/init.d/network restart
```

怎么样, 有线和无线上网又都回来了吧!










注意, 有的人在网上贴出了他的完整配置文件/etc/config/network 和/etc/config/wireless, 如果你复制他的文件覆盖你的文件, 再修改用户名和密码, 可能会出问题, 因为不同路由器的硬件配置可能不同

相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

命令行 uci设置 OpenWrt router 模式拨号上网

-  Ubuntu下载OpenWrt for TP-LINK wr2543N trunk版固件
-  scp复制固件到OpenWrt路由器 /tmp目录
-  ssh登录OpenWrt路由器
-  sysupgrade升级固件并取消保留原来配置文件
-  OpenWrt uci命令行设置拨号上网:
-  OpenWrt uci命令行设置无线上网:
-  允许远程主机用ssh的方式登录路由器及设置时区
-  ssh登录OpenWrt相关高级设置(你可能暂时用不到)
-  启用新的网络和无线设置

命令行 uci设置OpenWrt ap模式上网参数

前面章节已经说过了router模式上网的设置方法，主要是设置OpenWrt路由器wan口的拨号上网参数

OpenWrt路由器工作在ap模式下时，自身不需要拨号上网了，设置稍有不同

什么时候需要用到OpenWrt ap模式上网

光纤包月或包年上网时，通信公司一般会给你一个猫，如果猫里没有设置自动拨号上网，那么OpenWrt路由器就要用router模式，我们自己手动在OpenWrt里设置拨号上网

如果猫里已经设置好拨号上网，从猫的lan拉出一根网线插到电脑的网线接口，电脑直接可以上网了，再把这根网线插到路由器上，这时OpenWrt路由器就要设置成ap模式

如果是公司里或者家里有多个路由器，上级路由器里拉出一根网线插到电脑网线接口，电脑直接可以上网了，再把这根网线插到路由器上，这时OpenWrt路由器作为下级路由器，需要设置成ap模式

OpenWrt路由器AP模式网络设置

- 从光猫或上级路由器拉出网线，插到OpenWrt路由器的lan口(注意不是插到wan口)
- 命令行登录OpenWrt路由器，设置参数

假设光猫或上级路由器的IP地址是192.168.1.1，我们设置OpenWrt路由器的lan地址是 192.168.1.254，这也是登录OpenWrt路由器的地址

```
uci set network.lan.gateway=192.168.1.1
uci set network.lan.dns=192.168.1.1
uci set network.lan.ipaddr=192.168.1.254

uci set network.wan.proto=none

uci commit network

uci set dhcp.lan.ignore=1
uci commit dhcp

uci set wireless.@wifi-device[0].disabled=0
uci set wireless.@wifi-iface[0].mode='ap'
uci set wireless.@wifi-iface[0].ssid='eastking'
uci set wireless.@wifi-iface[0].encryption='psk2'
uci set wireless.@wifi-iface[0].key='icanfly9876'

uci commit wireless
wifi

/etc/init.d/network restart
```

客户端连接OpenWrt路由器：


如果不需要翻墙，客户端连上OpenWrt路由器后，直接就可以上网了

如果上级路由器没有翻墙，客户端需要通过OpenWrt路由器翻墙，客户端连接到OpenWrt路由器后，按照下面设置：

- 设置客户端连接的IPv4地址是 192.168.1.6(最后的6不和其他设备的地址相同即可)
- 设置子网掩码为255.255.255.0
- Router(网关)和DNS设为路由器lan口的地址，此处为192.168.1.254

原理：以OpenWrt路由器作为DNS服务器，我们已经把OpenWrt设置成翻墙路由器，连上的客户端自然就可以打败功夫网了

Auto-Join



IPV4 ADDRESS

Configure IP

Manual >

IP Address

192.168.1.6

Subnet Mask

255.255.255.0

Router

192.168.1.254

DNS

Configure DNS

Manual >

iPhone连接ap模式的翻墙路由器, IPV4设置如上图

Back

Configure DNS

Save

Automatic

Manual

DNS SERVERS

192.168.1.254

Add Server

iPhone连接ap模式的翻墙路由器, DNS设置如上图



相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name>

2018-12-07

命令行 uci设置OpenWrt ap模式上网参数

-  什么时候需要用到OpenWrt ap模式上网
-  OpenWrt路由器AP模式网络设置

37

OpenWrt 国内镜像源下载固件

从国内下载 OpenWrt 官方仓库的软件会比较慢, 解决办法是使用国内镜像

我们来调整一下 OpenWrt 存储库的设置

OpenWrt Feed 在 `/etc/opkg/distfeeds.conf` 中设置

```
kige@openwrt:~# cd /etc/opkg
kige@openwrt:/etc/opkg# ls
customfeeds.conf  distfeeds.conf  keys

kige@openwrt:/etc/opkg# cat dist*
src/gz openwrt_core http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/nand/packages
src/gz openwrt_base http://downloads.openwrt.org/releases/18.06.1/packages/mips_24kc/base
src/gz openwrt_luci http://downloads.openwrt.org/releases/18.06.1/packages/mips_24kc/luci
src/gz openwrt_packages http://downloads.openwrt.org/releases/18.06.1/packages/mips_24kc/packages
src/gz openwrt_routing http://downloads.openwrt.org/releases/18.06.1/packages/mips_24kc/routing
src/gz openwrt_telephony http://downloads.openwrt.org/releases/18.06.1/packages/mips_24kc/telephony
```

我们更换成中科大的镜像, 地址是: <http://openwrt.proxy.ustclug.org>

```
kige@openwrt:/etc/opkg# cp dist* distfeeds.conf.bak
kige@openwrt:/etc/opkg# sed -i s/downloads.openwrt.org/openwrt.proxy.ustclug.org/ /etc/opkg/distfeeds.conf
```

接下来还要做二件事:

- 把中科大镜像站域名加入 `/etc/dnsmasq.d/custom.china.conf`

直接修改 `accelerated-domains.china.conf` 不是好主意, 更新文件时我们的修改会被覆盖。`custom.china.conf` 包含了自定义的在国内DNS的域名

- 把中科大 OpenWrt 镜像的IP地址 `202.141.178.13` 加入 `/etc/shadowsocks-libev/ip_custom.txt`

`ip_custom.txt` 是自定义的路由器防火墙忽略的地址, 这样即使全局翻墙, 中科大OpenWrt镜像还是直连

如果你已经把本项目 clone 到了 Windows 下 C 盘根目录, 并且按照 [OpenWrt + Git Bash for Windows 快速切换翻墙模式](#) 设置好了一键切换翻墙模式, 那么你不用自己修改以上设置, 只要如下操作就行了:

调出 Git Bash for Windows, 执行命令:

```
MinGW64 ~$ cd /C/openwrt-fanqiang/openwrt/default
$ scp etc/dnsmasq.d/custom.china.conf router:/etc/dnsmasq.d/
$ scp etc/shadowsocks-libev/ip_custom.txt router:/etc/shadowsocks-libev/
$ ss-restart
$ feeds-cn
```

你可能猜到了 `ss-restart` 是自动登录路由器并重启 shadowsocks, 而 `feeds-cn` 则自动登录路由器并修改OpenWrt软件仓库 feeds 为国内镜像

这种直接在 OpenWrt 里执行命令的方法可比登录界面再去修改设置高效100倍。快点把你心爱的脚本分享出来, 并提交到 <https://github.com/softwaredownload/openwrt-fanqiang> 这样任何人都可以一键使用你的脚本而不用去研究技术细节了

相关资源:

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/bin>
- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/default/etc/dnsmasq.d>
- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/default/etc/shadowsocks-libev>
- <https://openwrt.org/docs/guide-user/additional-software/opkg>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

OpenWrt + shadowsocks-libev 实现路由器自动翻墙

相信经过前面的教程, 大家对OpenWrt和Linux Ubuntu有一定的熟悉了。如果还不熟悉Ubuntu, 就安装Ubuntu, 实际使用一个月前面的文章都是技术准备, 有基础的读者可以略过。在本章中, 我们要OpenWrt路由器安装 shadowsocks-libev来实践翻墙

最简单的路由器刷OpenWrt翻墙方案:

- <https://github.com/softwaredownload/openwrt-fanqiang>

在线阅读OpenWrt路由器翻墙、科学上网教程:

- <https://fanqiang.software-download.name>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-10-22

什么是shadowsocks-libev翻墙软件

shadowsocks-libev 是一个 shadowsocks 协议的轻量级实现, 是 shadowsocks-android, shadowsocks-ios 以及 shadowsocks-openwrt 的上游项目。它具有以下特点:

1. 体积小。静态编译并打包后只有 100 KB
2. 高并发。基于 libev 实现的异步 I/O, 以及基于线程池的异步 DNS, 同时连接数可上万
3. 低资源占用。几乎不占用 CPU 资源, 服务器端内存占用一般在 3MB 左右
4. 跨平台。适用于所有常见硬件平台, 已测试通过的包括 x86, ARM 和 MIPS。也适用于大部分 POSIX 的操作系统或平台, 包括 Linux, OS X 和 Cygwin 等
5. 协议及配置兼容。完全兼容 shadowsocks 协议, 且兼容标准实现中的 JSON 风格配置文件, 可与任意实现的 shadowsocks 客户端或服务端搭配使用

shadowsocks-libev 包括服务端和客户端两部分, 一共三个模块

1. ss-server: 服务端, 部署在远程服务器, 提供 shadowsocks 服务
2. ss-local: 客户端, 提供本地 socks5 协议代理
3. ss-redir: 客户端, 提供本地透明代理, 需要与 iptables NAT 表配合使用
4. ss-tunnel: 客户端, 本地端口转发

相关资源:

- <https://github.com/shadowsocks/shadowsocks-libev>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

翻墙软件Shadowsocks-libev服务端设置

要利用 shadowsocks-libev翻墙, 首先要有一台国外的服务器安装并运行shadowsocks 服务端。如果还没有服务器,可以到业界著名的 [Digital Ocean](#) 购买一台SSD虚拟服务器VPS, 全SSD硬盘, 速度极快

Ubuntu安装 shadowsocks-libev服务端

for Debian 9("Stretch"), unstable, Ubuntu 16.10 and later derivatives:

```
sudo apt-get update
sudo apt-get install shadowsocks-libev
```

for other versions:

```
#Add GPG public key:
wget -O- http://shadowsocks.org/debian/1D27208A.gpg | sudo apt-key add -

# Ubuntu 14.04 or above
sudo add-apt-repository "deb http://shadowsocks.org/ubuntu trusty main"

# Debian Wheezy, Ubuntu 12.04 or any distribution with libssl > 1.0.1
sudo add-apt-repository "deb http://shadowsocks.org/debian wheezy main"

sudo apt-get update
sudo apt-get install shadowsocks-libev
```

Ubuntu 16.10上确认shadowsocks-libev已经运行:

```
sudo systemctl status shadowsocks-libev
```

上述命令的效果:

- 安装ss-local ss-redir ss-server ss-tunnel...到 /usr/bin
- 启动文件 /etc/init.d/shadowsocks-libev
- 配置文件 /etc/shadowsocks-libev/config.json (旧版是/etc/shadowsocks/config.json)
- 一些默认启动配置 /etc/default/shadowsocks-libev (旧版是/etc/default/shadowsocks)

编辑shadowsocks-libev配置文件

```
sudo vi /etc/shadowsocks-libev/config.json
```

改成类似如下:

```
{
  "server": "[::0]", "0.0.0.0",
  "server_port": 1098,
  "password": "killgfw",
  "method": "chacha20-ietf-poly1305",
  "ipv6_first": true,
  "dns_ipv6": true,
  "fast_open": true,
  "timeout": 600
}
```

简要解释如下:

- "server": "[::0]", "0.0.0.0"
监听本机IPv6和IPv4地址
- "server_port": 1098
shadowsocks-libev 服务端 ss-server 监听的端口
- "password": "killgfw"

shadowsocks-libev客户端加密通信的密码, 有以下几个要求:

- shadowsocks服务端和客户端密码必须一致
- 密码长度不少于6位
- "method":"chacha20-ietf-poly1305"

加密算法, 详见 [Shadowsocks不同加密算法的区别](#)

- "fast_open":true

一种加速数据传送的优化, 必须要设置好才能启用这个选项。如果没有设置过, 值先改成 false

详见 [Ubuntu OpenWrt 开启 TCP Fast Open \(TFO\)流量加速](#)

防火墙 ufw 设置

ufw 是Ubuntu设置防火墙的工具, 查看 ufw 是否已经启用:

```
sudo systemctl status ufw
```

在 [Digital Ocean](#) 创建 VPS 后, 默认没有启用 ufw, 可以这样启用:

```
sudo ufw enable
```

启用了ufw以后, 那么要用如下命令开放server_port, 注意把下面的1098换成你的实际端口:

```
sudo ufw allow 1098
```

查看 ufw 状态

```
sudo ufw status
```

查看 ss-server 监听的端口:

```
netstat -lnp
```

你可以给 ss-server 启动参数加上或去掉 `-u` 运行 netstat 命令看看区别

给 shadowsocks-libev 创建 ufw profile

我们也可以换一种方式开放 1098 端口 给 shadowsocks-libev 服务端 ss-server

```
$ cd /etc/ufw/applications.d/
$ sudo vi shadowsocks

# add lines
[shadowsocks-libev]
title=shadowsocks-libev
description=shadowsocks-libev server
ports=1098/udp|1098/tcp
```

然后我们可以这样给shadowsocks-libev添加防火墙规则:

```
$ sudo ufw allow shadowsocks-libev
Rule added
Rule added (v6)

$ sudo ufw status verbose | grep 1098
1098/udp (shadowsocks-libev) ALLOW IN    Anywhere
1098/tcp (shadowsocks-libev) ALLOW IN    Anywhere
1098/udp (shadowsocks-libev (v6)) ALLOW IN    Anywhere (v6)
1098/tcp (shadowsocks-libev (v6)) ALLOW IN    Anywhere (v6)
```

更加清楚地显示了谁监听在什么端口

如果前面已经运行了 `sudo ufw allow 1098` 可以这样删除重复规则:

```
sudo ufw delete allow 1098
```

再用 netstat 命令查看一下 shadowsocks-libev 监听的端口：

```
$ sudo netstat -lnp | grep ss-server
tcp        0      0 0.0.0.0:1098          0.0.0.0:*        LISTEN      2414/ss-server
tcp6       0      0 :::1098              :::*              LISTEN      2414/ss-server
udp        0      0 0.0.0.0:1098          0.0.0.0:*        2414/ss-server
udp6       0      0 :::1098              :::*              2414/ss-server
```

控制shadowsocks-libev的方法

在Ubuntu 16.10上安装shadowsocks-libev后，默认已经随机启动了

```
sudo service shadowsocks-libev restart
sudo service shadowsocks-libev start
sudo service shadowsocks-libev stop
```

查看ss-server是否已经启动并且带有 -u启动参数

```
ps ax | grep ss-server
```

如果启动正常，返回结果类似如下：

```
/usr/bin/ss-server -c /etc/shadowsocks-libev/config.json -u
```

注意其中有-u。如果shadowsocks客户端启用了udp relay，而服务端启动时不带-u参数，翻墙自然就失败了

相关资源：

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/ubuntu/etc/shadowsocks-libev>
- <https://github.com/shadowsocks/shadowsocks-libev>
- <https://fanqiang.software-download.name/>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-12-07

翻墙软件Shadowsocks-libev服务端设置

-  Ubuntu安装 shadowsocks-libev服务端
-  编辑shadowsocks-libev配置文件
-  防火墙 ufw 设置
-  给 shadowsocks-libev 创建 ufw profile
-  控制shadowsocks-libev的方法
-  查看ss-server是否已经启动并且带有 -u启动参数

OpenWrt路由器运行 shadowsocks-libev ss-local 客户端

shadowsocks-libev for OpenWrt 要和 OpenWrt 版本一致, 否则可能无法安装, 或者安装了不能启动

shadowsocks-libev选择 OpenSSL 版还是 PolarSSL 版

根据依赖的 SSL 库可分为 OpenSSL 和 PolarSSL 两种版本OpenSSL 版依赖 libopenssl, 支持加密方式多, 体积大 PolarSSL 版依赖 libpolarssl, 体积小, 加密方式少

如果内存大就选OpenSSL版, 反之则选PolarSSL版

安装shadowsocks-libev客户端到OpenWrt路由器(星号替换成实际的字符)

```
~/Downloads$ scp shadowsocks-libev-polarssl_1.*_ar71xx.ipk root@192.168.1.1:/tmp/  
~/Downloads$ ssh root@192.168.1.1  
root@OpenWrt:~# cd /tmp  
root@OpenWrt:~# opkg install shadowsocks-libev-polarssl_1.*_ar71xx.ipk
```

修改shadowsocks-libev客户端配置

```
root@OpenWrt:~# vi /etc/shadowsocks-libev/config.json
```

改成类似如下:

```
{  
  "server": "1.0.9.8",  
  "server_port": 1098,  
  "local_port": 7654,  
  "password": "killgfw",  
  "method": "chacha20-ietf-poly1305"  
}
```

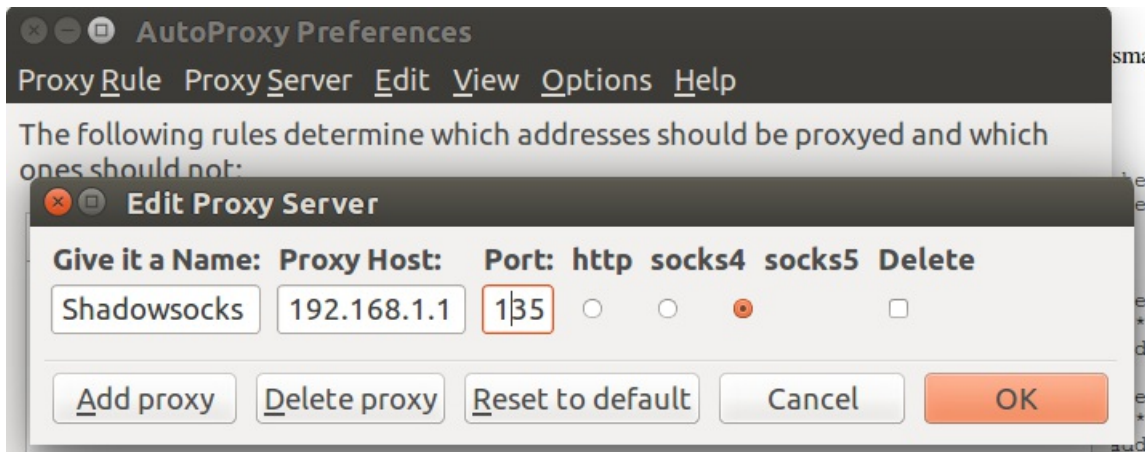
注意, server IP必须修改你的实际IP。其他可以保持默认

shadowsocks代理上网测试

- 启动shadowsocks 客户端:

```
root@OpenWrt:~# ss-local -c /etc/shadowsocks-libev/config.json
```

- Ubuntu浏览器代理上网设置, 以FireFox配合AutoProxy为例, 增加Proxy Server, Proxy Host填192.168.1.1,Port是7654, 勾选Sock5.如下图:



Ubuntu设置AutoProxy的默认代理是shadowsocks,就可以打开被墙的网站如[YouTube.com](https://www.youtube.com)

Windows 电脑使用 shadowsocks-libev 客户端 ss-local 翻墙的方法见下面链接:

<https://fanqiang.software-download.name/ebook/04.8.html>

以前我在每台电脑上都运行一个shadowsocks客户端, 每台电脑都要像上面这样配置浏览器代理上网翻墙。但是还是太复杂, 如果家里有十台上网设备, 所有要连国外网站的软件都可能要配置代理访问, 有些软件还根本没有设置代理的接口。有没有更简单的方法呢?

现在路由器里安装了shadowsocks, 所有有线和无线上网设备都不用分别安装shadowsocks了, 非常方便

相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

OpenWrt路由器运行 shadowsocks-libev ss-local 客户端

-  shadowsocks-libev选择 OpenSSL 版还是 PolarSSL 版
-  安装shadowsocks-libev客户端到OpenWrt路由器(星号替换成实际的字符)
-  修改shadowsocks-libev客户端配置
-  shadowsocks代理上网测试

史上最通俗易懂的OpenWrt翻墙路由器解释

什么是域名和IP地址

每个网站都可以有两个唯一标识:域名和IP地址。域名相当于人的名字, IP地址相当于该人使用的电话号码。(不同之处:域名是唯一的, 人的名字会有重名)

网站为什么要有两个标识?域名是为了方便人类记忆的, 比如YouTube.com, 而电脑处理却喜欢处理数字, 纯数字格式的IP地址就是为了让电脑查找计算方便些

通过域名查询IP的那些事情

我们在浏览器地址栏里输入 www.youtube.com 并回车, 到底会发生哪些不可思议的事情呢:

- 浏览器问就近的某台电脑(叫域名服务器):Hi, youtube.com的IP地址是什么?
- 域名服务器:不就是 74.125.239.98
- 浏览器:谢谢。我就到你给我的地址去找内容了

还有种情况, 浏览器第一次问的域名服务器不知道某域名的IP地址:

- 浏览器问就近的域名服务器:Hi, youtube.com的IP地址是什么?
- 域名服务器:这个我不知道哇, 我帮你问问我的上线
- 上线服务器:我也不知道哇, 我也只好问我的上线, 等等, 别挂掉
- 某域名服务器:这么简单还来问我, 不就是 74.125.239.98
- 浏览器:谢谢。我就到你给我的地址 74.125.239.98 去找内容

白脸很忙, 不看YouTube(看不懂?)

在中国, YouTube为什么被封?YouTube有几千万, 上亿个视频, 如果某几个视频让某些人看了不爽, 就来个宁可错杀百万, 不可放过一个, 把整个YouTube给封了, 全国人民都无法正常访问YouTube了

这个时候, 又发生了哪些不可告人的事情呢?

1. 浏览器问就近的域名服务器:喂, youtube.com的IP地址是什么?
2. 中国的某域名服务器:这我知道, 44.44.44.44, (心里嘀咕, 我给你的是太平洋海底的地址, 你能找到内容才怪呢, 白脸(领导)很忙, 天朝很好, 访问这种破网站干啥, 满屏洋文, 我怎么看得懂, 哼)
3. 浏览器:谢谢。我这就去找主人需要的内容。。。找了好久, 还是什么也没找到,我的命怎么这么苦。。

阳光底下, 每时每刻每秒, 这样齷齪的事情在发生千次, 万次, 亿次...

深刻理解 TCP UDP 通信协议

UDP是什么意思:

UDP 是User Datagram Protocol的简称, 中文名是用户数据报协议, 是OSI(Open System Interconnection, 开放式系统互联)参考模型中一种无连接的传输层协议, 提供面向事务的简单不可靠信息传送服务, IETF RFC 768是UDP的正式规范。UDP在IP报文的协议号是17 UDP协议全称是用户数据报协议[1], 在网络中它与TCP协议一样用于处理数据包, 是一种无连接的协议。在OSI模型中, 在第四层——传输层, 处于IP协议的上一层。UDP有不提供数据包分组、组装和不能对数据包进行排序的缺点, 也就是说, 当报文发送之后, 是无法得知其是否安全完整到达的。UDP用来支持那些需要在计算机之间传输数据的网络应用。包括网络视频会议系统在内的众多的客户/服务器模式的网络应用都需要使用UDP协议。UDP协议从问世至今已经被使用了很多年, 虽然其最初的光彩已经被一些类似协议所掩盖, 但是即使是在今天UDP仍然不失为一项非常实用和可行的网络传输层协议 与所熟知的TCP(传输控制协议)协议一样, UDP协议直接位于IP(网际协议)协议的顶层。根据OSI(开放系统互连)参考模型, UDP和TCP都属于传输层协议。UDP协议的主要作用是将网络数据流量压缩成数据包的形式。一个典型的数据包就是一个二进制数据的传输单位。每一个数据包的前8个字节用来包含报头信息, 剩余字节则用来包含具体的传输数据

缺点就是优点, UDP通信效率高:

既然 UDP 数据发送之后, 是无法得知其是否安全完整到达的, 那么为什么 在 shadowsocks 中还要用 UDP 呢?不用校验数据是否完整, 数据传递的速度自然更快。所以在游戏界, 基于 UDP 协议的网络通信又被称作高性能网络。联机游戏要在服务端和客户端之间传递大量数据, 对通信效率要求很高, 因此多用 UDP

UDP的数据可能不完整, 这限制了 UDP 协议的用途, 更多的地方用的是 TCP 协议。但是也有例外, QQ 就是采用 UDP 协议通信的。一般来说即时通信适合用 TCP, 腾讯在 UDP 的基础上进行了高度的封装、优化, 使之一定程度兼具 TCP UDP 两者的优点

UDP 最常见的用途是 DNS 查询。我们打开一个网页, 会有多次的 DNS 查询动作, 在进行 DNS 查询的时候, 通信流量默认就是走 UDP 协议。DNS 规范中包含了 TCP 协议, 但是 TCP 只是一种备选方案, 很多公共 DNS 查询提供商并不提供 TCP 查询的接口

Shadowsocks 是一个优秀的 Socks 代理工具, 在很长的一段时间里它仅支持 TCP 代理, 后来在 Shadowsocks-libev 上实现了 UDP 转发的功能, 然后我们才能在 shadowsocks 客户端把 DNS 查询请求转发到 shadowsocks 服务端, 由服务端把查询到的数据返回到客户端, 这就避免了 GFW 的域名污染

shadowsocks 如果使用 TCP 协议转发域名查询请求到服务端, 客户端和服务端的通信会被 GFW 直接重置

太阳要升起, 网民要雄起

但是, 还有问题没有解决:

网站有两种, 国内的和国外的。如果不分国内国外全部都到国外去查询域名的IP, 访问国内的网站就会变慢。虽然有心逃离, 还是无法割断哪

有几种解决方案:

1. 建国外重要网站名单, 简称外单(黑名单, gfwlist), 外单上的域名都到国外去查询IP, 其他就在国内查询

如果IP地址在外单上, 就加密访问, 领导不知道我访问了这个地址, 这样领导的心情可能会好些

2. 同样是建立外单。不同的是, 我不想花费精力去区分某个IP是不是在外单上, IP地址可能经常在变, 这样做不怕累吗。我的办法是, 不是中国的IP, 全部加密访问

3. 每个人的用途不同, 谁有本事建立通用的外单?

即使有人建立了包含很多域名的外单, 网站内容往往是互相引用的, 某外单上网站引用了不在外单上的被封网站, 导致打网站贼慢, 这个该怎么办? 难道要手动查看网页源代码, 一个个地搜索查找, 逐个测试?

最简单有交的方法, 是给国内重要网站建立名单, 简称内单。内单上的网站都在国内dns, 其他网站全部到国外dns。访问非中国的IP都流量加密

我曾经用过第一种方案, 试图用网友整理的外单(ChinaDNS), 但是, 在实际使用过程中, 经常需要临时增加外单域名并重启路由器, 有时一天要重复好多次, 不胜其烦。

第三种方案, 就是本教程使用的方案, 是目前来说比较好的方案

OpenWrt翻墙路由器内部发生的故事(千万别告诉白脸):




1. 浏览器:喂, 谁知道YouTube.com的IP, 主人要用
2. 路由器:稍等, 我查下主人设置的内单, 稍等。。。不在内单, 我通过秘密通道查
3. 浏览器:喂, 告诉我baidu.com的IP
4. 路由器:哇, 内单, 马上就给你
5. 浏览器:请给我IP地址60.188.5.6的内容
6. 路由器:等下, 立即就好。。。中国IP, 该那就那去取内容。不是中国IP, 借道主人的秘密通道去取内容

相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

[史上最通俗易懂的OpenWrt翻墙路由器解释](#)

-  什么是域名和IP地址
-  通过域名查询IP的那些事情
-  白脸很忙, 不看YouTube(看不懂?)
-  深刻理解 TCP UDP 通信协议
-  太阳要升起, 网民要雄起

配置 OpenWrt shadowsocks 路由器智能自动翻墙

OpenWrt路由器用dnsmasq转发国内重要域名查询

OpenWrt默认自带dnsmasq, 我们只要配置一下就好了。ssh登录OpenWrt路由器后:

- 建立dnsmasq.d目录:

```
root@OpenWrt:~# mkdir /etc/dnsmasq.d
root@OpenWrt:~# echo "conf-dir=/etc/dnsmasq.d" >> /etc/dnsmasq.conf
```

- OpenWrt安装GNU wget以支持https下载, 下载国内重要网站名单, 用国内域名服务器查询IP地址

```
root@OpenWrt:~# cd /etc/dnsmasq.d
root@OpenWrt:/etc/dnsmasq.d# opkg install wget
root@OpenWrt:/etc/dnsmasq.d# wget -4 --no-check-certificate -O /etc/dnsmasq.d/accelerated-domains.china.conf https://github.com/felixonmars/dnsmasq-china-list/raw/master/accelerated-domains.china.conf
root@OpenWrt:/etc/dnsmasq.d# wget -4 --no-check-certificate -O /etc/dnsmasq.d/bogus-nxdomain.china.conf https://github.com/felixonmars/dnsmasq-china-list/raw/master/bogus-nxdomain.china.conf
```

注: [accelerated-domains.china.conf](https://github.com/felixonmars/dnsmasq-china-list/raw/master/accelerated-domains.china.conf) 文件中的条目举例:

```
server=/10010.com/114.114.114.114
server=/115.com/114.114.114.114
```

意思是, 访问10010.com这个结尾的域名时, dnsmasq会转发到国内的域名服务器114.114.114.114进行dns查询

gfwlist.conf: 其他域名, 转发到shadowsocks-libev ss-tunnel指定的端口dns查询

```
root@OpenWrt:/etc/dnsmasq.d# echo "server=/#/127.0.0.1#3210" > gfwlist.conf
```

上面 # 是通配符, 代表泛匹配所有域名。dnsmasq匹配域名的特点是详细特征优先匹配, 因此会先匹配accelerated-domains.china.conf 上的域名, 如果不匹配, 再匹配这条规则: 转发到本地端口3210进行域名查询

后面我们会配置shadowsocks-libev的本地客户端ss-tunnel转发本地端口3210的查询到远程自建服务器

配置shadowsocks本地客户端ss-redir启动和停止函数

```
root@OpenWrt:/etc/dnsmasq.d# vi /etc/init.d/shadowsocks
```

[/etc/init.d/shadowsocks](#):

```
#!/bin/sh /etc/rc.common

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Last Update: 2018-09-27

START=95

SERVICE_USE_PID=1
SERVICE_WRITE_PID=1
SERVICE_DAEMONIZE=1

start() {
    echo 'server=/#/127.0.0.1#3210' > /etc/dnsmasq.d/gfwlist.conf
    /etc/init.d/dnsmasq restart

    service_start /usr/bin/ss-redir -b 0.0.0.0 -c /etc/shadowsocks-libev/config.json -f /var/run/shadowsocks.pid -u
    service_start /usr/bin/ss-tunnel -b 0.0.0.0 -c /etc/shadowsocks-libev/config.json -l 3210 -L 8.8.4.4:53 -u
    /usr/bin/ss-firewall-asia
    #/usr/bin/ss-firewall-global
    #/usr/bin/ss-firewall-china
}
```



```

}

stop() {
    echo 'server=/#/114.114.114.114' > /etc/dnsmasq.d/gfwlist.conf
    /etc/init.d/dnsmasq restart

    service_stop /usr/bin/ss-redir
    service_stop /usr/bin/ss-tunnel
    service_stop /usr/bin/obfs-local
    killall ss-redir
    killall ss-tunnel
    killall obfs-local
    /etc/init.d/firewall restart
}

```

shadowsocks本地客户端配置文件start stop函数说明:

- `echo 'server=/#/114.114.114.114' > /etc/dnsmasq.d/gfwlist.conf`

停止shadowsocks翻墙服务时,要把泛匹配域名的解析转发到国内的dns服务器,这里是114

原来用的是 sed 替换字符串的方法, 有几次发现 gfwlist.conf 被意外清空, 导致翻墙失败, 原因不理。现改用 echo 清空文件并添加字符串的方法后, 即使 gfwlist.conf 内容被清除, 也能在重启 shadowsocks 后重新写入

- `echo 'server=/#/114.114.114.114' > /etc/dnsmasq.d/gfwlist.conf`

开启翻墙服务时, 如果以前停止过shadowsocks翻墙服务,确保泛匹配域名的解析通过ss-tunnel 3210端口转发

- `service_start /usr/bin/ss-tunnel -b 0.0.0.0 -c /etc/shadowsocks-libev/config.json -l 3210 -L 8.8.4.4:53 -u`

监听本地3210端口, 转发到自己的服务器的53端口向8.8.4.4查询DNS

- `/usr/bin/ss-firewall-...`

dnsmasq只是负责域名查询分配转发, 查询到IP地址后, 是否需要通过shadowsocks加密请求内容, 要在ss-firewall-...里进行设置

三个 `/usr/bin/ss-firewall-...` 启用其中一个

浏览外网时建议启用 `/usr/bin/ss-firewall-global` 全局翻墙, 既节省了路由器的计算资源, 又避免一些意想不到的问题

- 运行 `/etc/init.d/shadowsocks stop` 有时并没有结束ss-redir 或ss-tunnel进程

这会导致修改 shadowsocks.conf 后需要重启路由器才能生效。加上 `killall` 强制杀掉进程避免重启。(2016-01-19)

(注:即使加了 `killall`, 有时还是不能杀掉进程, 这种情况就只能重启路由器了。也就是说, 修改了翻墙配置, 有时必须重启路由器才能生效)

- ss-redir 加上 `-u` 参数

据tefiszx的建议:

- youtube目前使用quic, udp协议是首选, 目前路由器的ss配合iptables只能转发tcp流量, 最新版的youtube的app会出现断流, 而网页端播放目前没问题(chrome必须关闭quic)
- 很多游戏的网络版都使用udp协议。目前的转发存在一定的局限性
- 在转发tcp流量的基础上增加转发upd后, ss就成为准vpn了。应用面更广
- 已经找到youtube安卓客户端3秒断流的解决办法。方法如下:
 - 修改/etc/init.d/shadowsocks文件, 把ss-redir 加-u参数启动
 - 修改/usr/bin/ss-firewall-*文件, 在防火墙规则中增加一条iptables规则, 将443端口的upd重定向到shadowsocks监听的端口即可

配置iptables防火墙转发IP和端口

```

root@OpenWrt:~# cd /usr/bin
root@OpenWrt:~# touch ss-firewall-asia
root@OpenWrt:~# chmod +x ss-firewall-asia
root@OpenWrt:~# vi ss-firewall-asia

```

`/usr/bin/shdowsocks-firewall-asia:`

```

#!/bin/sh

# Author:      https://github.com/softwaredownload/openwrt-fanqiang
#             phoeagon tefiszx idonknown
# Last Update: 2018-10

#create new chains

```

```

iptables -t nat -N SHADOWSOCKS
iptables -t nat -N SHADOWSOCKS_WHITELIST

# Ignore your shadowsocks server-s's addresses
# It's very IMPORTANT, just be careful
# you'd better add them in an individual file
for white_ip in `cat /etc/shadowsocks-libev/ip_server.txt`;
do
    iptables -t nat -A SHADOWSOCKS -d "${white_ip}" -j RETURN
done

# Ignore Custom IP list
for white_ip in `cat /etc/shadowsocks-libev/ip_custom.txt`;
do
    iptables -t nat -A SHADOWSOCKS -d "${white_ip}" -j RETURN
done

# for Chrome youtube
iptables -t nat -A SHADOWSOCKS -p udp --dport 443 -j REDIRECT --to-ports 7654

# Ignore LANs to bypass the proxy
# See Wikipedia and RFC5735 for full list of reserved networks.
iptables -t nat -A SHADOWSOCKS -d 0.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 10.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 127.0.0.0/8 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 169.254.0.0/16 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 172.16.0.0/12 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 192.168.0.0/16 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 224.0.0.0/4 -j RETURN
iptables -t nat -A SHADOWSOCKS -d 240.0.0.0/4 -j RETURN

# Check whitelist
iptables -t nat -A SHADOWSOCKS -j SHADOWSOCKS_WHITELIST
iptables -t nat -A SHADOWSOCKS -m mark --mark 1 -j RETURN

# Anything else TCP request should be redirected to shadowsocks's local port
iptables -t nat -A SHADOWSOCKS -p tcp -j REDIRECT --to-ports 7654
# Apply the rules
iptables -t nat -A PREROUTING -p tcp -j SHADOWSOCKS

# Or ignore Asia IP address
for white_ip in `cat /etc/shadowsocks-libev/ip_asia.txt`;
do
    iptables -t nat -A SHADOWSOCKS_WHITELIST -d "${white_ip}" -j MARK --set-mark 1
done

# Ignore China IP address
# See ashi009/bestrouetb for a highly optimized CHN route list.
#for white_ip in `cat /etc/shadowsocks-libev/ip_china.txt`;
#do
#    iptables -t nat -A SHADOWSOCKS_WHITELIST -d "${white_ip}" -j MARK --set-mark 1
#done

```

OpenWrt路由器 iptables防火墙设置含义:

- [/etc/shadowsocks-libev/ip_server.txt](#)

如果本地发出请求到shadowsocks服务端 IP,就返回, 不作任何特殊处理

2018-10 起, 防火墙规则中要忽略的 IP 列表分类保存到 /etc/shadowsocks-libev/ 目录下, 这给我们修改带来了很大的便利。特别是你有多个 VPS时, 全写在 ip_server.txt 就行了

确保这几个 ip_*.txt 路由器 /etc/shadowsocks-libev 保存在目录下, 否则会出错。可以到下面地址下载这几个文件:

<https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/default/etc/shadowsocks-libev>

- [/etc/shadowsocks-libev/ip_custom.txt](#)

自定义的忽略 IP, 这个文件很有必要, 比如你在电脑WIFI连接的属性中设置一个国内某个特殊DNS地址, 再把这个地上加到ip_custom.txt, DNS解析速度会非常快

默认启用的是 ss-firewall-asia, 即亚洲 IP 加入白名单, 这样防火墙规则简单多了, 但是有些 .tw .hk 网站可能打不开, 把它们的 ip 加到 ip_custom.txt 就可以打开了

- 如果本地发出请求到局域网, 也立即返回

本来计划启用 ip_lan.txt, 但这样做有风险, 如果你忘记在路由器里放置该文件, 那么将无法登录路由器, 只能重新刷固件

- [/etc/shadowsocks-libev/ip_asia.txt](#)

如果发出请求到亚洲的IP地址, 也立即返回

- 剩下的IP内容请求, 全部转发到shadowsocks-libev本地客户端ss-redir监听的端口, 由ss-redir负责和服务端进行加密通讯。(手下报告访问youtube的屁民为个位数, 白脸心里那个高兴啊。可惜经过加密, 内容传输速度会有下降)

- iptables -t nat -N SHADOWSOCKS_WHITELIST 相关行

首先运行全代理模式, 然后再执行白名单。在白名单比较长时路由器冷启动的速度会比较快, 如果启用了ip_china.txt, 你会感觉到路由器启动速度快了好多倍。(Thanks Phoeagon)

一般不建议使用 [ip_china.txt](#)

预编译翻墙固件都带了这个文件, 这个文件很长, 因此配置不高的路由器DIR-505, 预编译固件里应该“发出请求到亚洲的IP地址就立即返回”

每个请求都检测是否中国区IP, 可能对路由器的压力较大, WNDR4300 路由器也是如此

从 2018-09 开始, 默认配置启用 Ignore Asia IP address, 这样路由器的压力就小得多了

如果你的路由器性能较好, 可以手动启用 Ignore China IP address, 并把Ignore Asia IP address段注释掉

- iptables -t nat -A SHADOWSOCKS -p udp --dport 443 -j REDIRECT --to-ports 7654

据tefisz建议, udp协议 443端口流量转发到shadowsocks 的本地端口, chrome 浏览器打开 youtube 可能更快

OpenWrt路由器防火墙设置重要说明:

- 你必须把上面的1.0.9.8换成你服务器真实的IP地址
- iptables -t nat -A SHADOWSOCKS -p tcp -j REDIRECT --to-ports 7654 这里的 7654 必须和OpenWrt路由器 `/etc/shadowsocks-libev/config.json` 里的 `local_port` 一样, 也就是说, 如果 `/etc/shadowsocks-libev/config.json` 里 `local_port":1090` 那这里的 7654 也要改成 1090

控制shadowsocks本地客户端的方法

```
root@OpenWrt:~# /etc/init.d/shadowsocks start
root@OpenWrt:~# /etc/init.d/shadowsocks enable
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks disable
```

说明:

- enable: 设置shadowsocks在OpenWrt路由器启动时自动启动
- start: 运行shadowsocks
- stop: 停止shadowsocks
- disable: 取消shadowsocks随机启动

启动并测试shadowsocks-libev本地客户端

确保所有设置无误后, 可以启动测试一下:

```
root@OpenWrt:~# /etc/init.d/dnsmasq restart
root@OpenWrt:~# /etc/init.d/shadowsocks enable
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks start
```

然后在Ubuntu电脑, 手机等设备上打开[youtube.com](#), [twitter.com](#)

下载配置文件的最新版

```
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

git clone 项目到本地后, 可以进入 openwrt目录查看文件

如果所有设置都正确, 应该可以较快速度打开被墙网站

相关资源:

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/default/etc/shadowsocks-libev>
- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/default/usr/bin>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

配置 OpenWrt shadowsocks 路由器智能自动翻墙

- 📄 OpenWrt路由器用dnsmasq转发国内重要域名查询
- 😊 gfwlist.conf: 其他域名, 转发到shadowsocks-libev ss-tunnel指定的端口dns查询
- 🔄 配置shadowsocks本地客户端ss-redir启动和停止函数
- 🛡️ 配置iptables防火墙转发IP和端口
- 🐱 控制shadowsocks本地客户端的方法
- ☁️ 启动并测试shadowsocks-libev本地客户端
- 📄 下载配置文件的最新版

OpenWrt自动更新设置和屏蔽广告

OpenWrt路由器自动更新国内重要网站名单

登录路由器后：

```
root@OpenWrt:~# cd /usr/bin
root@OpenWrt:~# touch chinalist
root@OpenWrt:~# chmod +x chinalist
root@OpenWrt:~# vi chinalist
```

[/usr/bin/chinalist](#):

```
#!/bin/sh

wget -4 --no-check-certificate -O /etc/dnsmasq.d/accelerated-domains.china.conf https://github.com/felixonmars/dnsmasq-china-list/raw/master/accelerated-domains.china.conf
wget -4 --no-check-certificate -O /etc/dnsmasq.d/bogus-nxdomain.china.conf https://github.com/felixonmars/dnsmasq-china-list/raw/master/bogus-nxdomain.china.conf
wget -4 --no-check-certificate -O /etc/dnsmasq.d/apple.china.conf https://github.com/felixonmars/dnsmasq-china-list/raw/master/apple.china.conf
wget -4 --no-check-certificate -O /etc/dnsmasq.d/google.china.conf https://github.com/felixonmars/dnsmasq-china-list/raw/master/google.china.conf
```

accelerated-domains.china.conf 内容越来越多了，可能会影响路由器的运行速度。这个文件其实应该分成二个：

— accelerated-domains.china.master.conf — accelerated-domains.china.slave.conf

master.conf 包含国内权重最高的几千个网站，每个路由器可以使用这个文件 slave.conf 包含国内权重不太高的网站，性能不高的路由器可以不使用这个文件

如果你的路由器性能不高，建议不要经常更新accelerated-domains.china.conf 以免路由器速度越来越慢。默认配置不再更新此文件，如果你的路由器性能较好，你可以手动更新

非特殊情况，可以不用 accelerated-domains.china.conf，有的国内 DNS 可以同时解析国内外网站。路由器里只要保留广告屏蔽就行了

OpenWrt路由器自动屏蔽广告

[/etc/dnsmasq.d](#)下有个 [ad-cn.conf](#) 文件，内容类似如下：

```
server=/.mobads.baidu.com/127.0.0.0
server=/.mobads-logs.baidu.com/127.0.0.0
server=/.media.admob.com/127.0.0.0
...
```

意思是.mobads.baidu.com的域名解析转发到 127.0.0.0，这个地址不具备域名解析的功能，于是就达到了屏蔽广告的功能

运行命令：

```
root@OpenWrt:~# cd /usr/bin
root@OpenWrt:~# touch blockad-cn
root@OpenWrt:~# chmod +x blockad-cn
root@OpenWrt:~# vi blockad-cn
```

[/usr/bin/blockad-cn](#):

```
#!/bin/sh

# Author:      https://github.com/softwaredownload/openwrt-fanqiang
# last update: 2018-10

TMP_HOSTS=/tmp/block.hosts.unsorted
HOSTS=/etc/dnsmasq.d/ad-cn.conf

# remove any old TMP_HOSTS that might have stuck around
rm ${TMP_HOSTS} 2> /dev/null

for URL in \
    "https://github.com/softwaredownload/cnhosts/raw/data/_build/tmp/full/hosts" \
    "https://github.com/e32ubhds/Hosts/raw/master/Hosts"
```

```
do
    # filter out comment lines, empty lines, localhost...
    # remove trailing comments, space( ,tab), empty line
    # replace line to dnsmasq format
    # remove carriage returns
    # append the results to TMP_HOSTS
    wget -4 --no-check-certificate -qO- "${URL}" | grep -v -e "^#" -e "^s*$" -e "localhost" -e "broadcasthost" -e "ip6" -e "^;" -e "^@" -e "^:" -e
    "^[a-zA-Z]" \
    | sed -E -e "s/#.*$//" -e "s/[[:space:]]*/ /g" -e "/^$/d" \
    -e "s/^127.0.0.1/server=\/.\/" -e "s/0.0.0.0/server=\/.\/" -e "/^[0-9].*/d" -e "s$/\/127.0.0.0/" \
    | tr -d "\r" >> ${TMP_HOSTS}

done

# remove duplicate hosts and save the real hosts file
sort ${TMP_HOSTS} | uniq > ${HOSTS}

rm ${TMP_HOSTS} 2> /dev/null
```

OpenWrt自动生成广告屏蔽列表说明：

- 2018-10起, blockad分成 blockad-cn 和 blockad-en 分别用于屏蔽国内外广告
- 运行上面命令产生的广告屏蔽列表比较长, 如果路由器性能比较低, dnsmasq匹配域名负荷会太大
- 如果dnsmasq超负荷工作, 可能会失去响应, 导致打不开网页, 这时需要登录路由器运行命令：
/etc/init.d/dnsmasq restart
- 所以, 还是尽量用性能好点的路由器吧

路由器性能比电脑差很多, 如果屏蔽列表很长, 那么短时间内快速打开数个网页就可能导致dnsmasq失去响应。最好是看完一个网页就关闭一个, 再打开新的网页

我认为在多数情况下, 屏蔽广告可以在电脑里操作, 移动设备可以用专门的广告屏蔽软件。当然, 如果路由器性能很是强悍, 在路由器里屏蔽广告是最爽的事情

通常的做法, 在路由器里屏蔽部分域名, 然后在电脑里设置更广泛、精确的屏蔽, 主要是设置host文件屏蔽和浏览器插件屏蔽

浏览器插件屏蔽, 可以装这些Chrome浏览器插件: uBlock Origin, Adfree.Player.Online。其中uBlock Origin的作用和Adblock Plus类似, 但是设置更加丰富

🕒 计划任务: 定时更新dnsmasq配置文件和自动重启shadowsocks

```
root@OpenWrt:~# crontab -e
```

输入以下内容：

```
* /30 * * * * isfound=$(ps | grep "ss-redir" | grep -v "grep"); if [ -z "$isfound" ]; then echo "$(date): restart ss-redir...">>/tmp/log/ss-monitor.log && /etc/init.d/shadowsocks restart; fi
* 12 * * * /usr/bin/chinalist
* 12 * * * /usr/bin/blockad-cn
```

OpenWrt计划任务说明：

- 每半小时检查shadowsocks-libev 客户端, 如果退出就自动重启
- 每天中午12点运行chinalist
- 每天中午12点运行blockad-cn

2014-09-24版的dir505, wr2543预编译固件是启用了计划任务的, 这会有潜在的不确定性, 如果更新时下载的文件如accelerated-domains.china.conf存在错误, 导致dnsmasq无法启动, 翻墙功能自然失效

如果你启用了上面的计划任务, 某一天突然不能翻墙了, 这时设置客户端的IP地址为和路由器同网段, 登录路由器, 用ps命令查看dnsmasq进程是否启动了, 如果没有启动, 就重刷固件或者用 <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/default/etc/dnsmasq.d>

下面的文件代替路由器里/etc/dnsmasq.d/下的文件

一般不建议自动更新 /etc/dnsmasq.d/的文件, 以免给翻墙失败时排查原因增加难度。可以手动运行命令更新, 更新后立即重启 dnsmasq 测试一下上网是否正常

附录: 计划任务定时关闭路由器OpenWrt:

人类的本性是目光短浅，玩得一时兴趣就会忘记定时休息的重要性。解决办法是在路由器里设置计划任务，禁止夜里某个时间段里使用路由器。下面的例子中，每20分钟检测一次，如果迟于20点10分或者早于7点就自动关闭OpenWrt路由器。这对小孩子特别有用，现在很多孩子使用电子设备上瘾，一个人睡的话甚至半夜在被窝里偷偷上网，现在好了，除非孩子强大到会登陆路由器修改设置，否则半夜重启路由器都无法通过路由器上网了

不过现在移动流量越来越便宜了，路由器自动关机控制的主要是我们自己使用电脑的时间，而不是控制小孩玩移动设备了




```
* /20 * * * * TIME=$(date +"%H%M"); if [ $TIME -ge 2010 ] || [ $TIME -le 700 ]; then poweroff; fi
```

相关资源：

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/default/usr/bin>
- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/default/etc/dnsmasq.d>
- <https://github.com/felixonmars/dnsmasq-china-list>
- <https://fanqiang.software-download.name/>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-12-07

[OpenWrt自动更新设置和屏蔽广告](#)

-  [OpenWrt路由器自动更新国内重要网站名单](#)
-  [OpenWrt路由器自动屏蔽广告](#)
-  [计划任务:定时更新dnsmasq配置文件和自动重启shadowsocks](#)

OpenWrt路由器为什么会翻墙失败或不稳定

给路由器刷上OpenWrt, 并按照 [本教程](#) 设置了服务端和客户端, 但还是有问题, 怎么办?



有的网站能翻墙, 有的网站翻墙失败或者打开非常慢, 这是怎么回事

- This site can't be reached
gfw.com unexpectedly closed the connection.
ERR_CONNECTION_CLOSED
- This site can't be reached
The connection was reset
ERR_CONNECTION_CLOSED
- This site can't be reached
www.tumblr.com took too long to respond
ERR_TIMED_OUT

有不少朋友遇到过上边这些情况, 打开 <https://google.com> 或者 <https://youtube.com> 没有问题, 但是有的网站很难打开, 或者干脆打不开

比如 <https://tumblr.com> 或者 <https://flickr.com> 有时就很难打开

这是怎么回事, 百思不得其解, 如果说翻墙配置不对, 又怎么解释有的网站翻墙没有问题呢

难道电脑系统设置有问题? 查找各种教程, 一大通修改后, 还是不行

再换用不同的浏览器, Chrome, FireFox, Edge, IE, 不行的还是不行

难道 GFW 成精了, 不是说好了建国后不能成精的吗

最后的努力, 不成功就向 妖精 投降算了:

- 更换成最好的加密方式
- 加上流量混淆插件
- 更换服务端IP地址、端口
- 软件都手动编译到最新版

忙活了几天, 终于升级完成, 一测试, 不行的还是不行, 忍不住, 胸膛起伏不定, 犹如万马奔腾...

且慢向妖精投降! 妖精最终总是被收服的, 或收到后宫, 或收作坐骑

大道至简, 不区分国内、国外 IP 地址, 全局翻墙, 所有流量都走 shadowsocks 加密通道往往能解决这个问题

现在的网页往往比较复杂, 会引用各种外部组件, 这些外部组件可能是属于不同的 IP 地址, 还有 CDN 动态加速, 有时可能会出现十分复杂的情况

比如你访问一个著名网站, 网站必定会判断你的区域, 首先根据你的 IP 地址定位你在 New York

网站有许多组件, 有的组件可能根据你的计算机特征, 比如系统语言判断你的区域, 于是可能判断你的区域为中国, 于是到中国的CDN取内容给你...数据在全球来回打转, 不但把路由器拖累了, 也把网站搞糊涂了, 导致网站打开的速度非常慢, 或者打不开

我们在路由器里区分国内 IP 和 国外 IP 的本意是为了打开网页更快, 然而, 有时却适得其反, 甚至导致网页根本打不开

为了方便在需要时切换不同的翻墙方式, 2018年9月起, `/usr/bin/ss-firewall-asia` 分成三个文件:

- `/usr/bin/ss-firewall-global`
- `/usr/bin/ss-firewall-china`
- `/usr/bin/ss-firewall-asia`

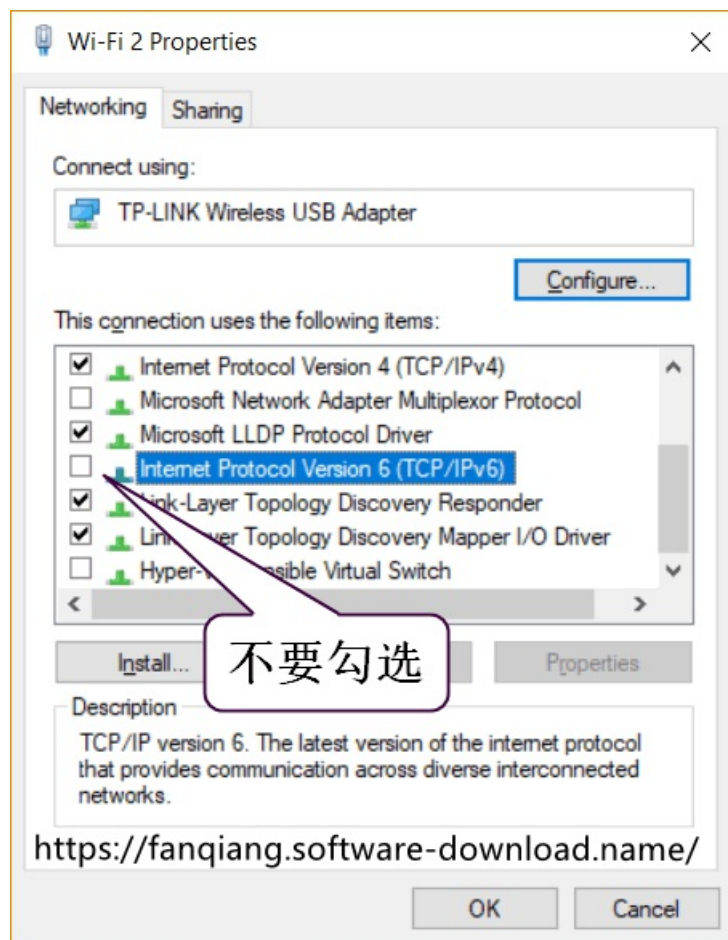
分别是: 默认, 全局翻墙, 忽略中国IP, 忽略亚洲IP



极少数网站比如 www.dropbox.com 打不开, 怎么办

www.dropbox.com 服务器默认使用IPv6地址, DropboBox Windows 电脑客户端也默认连接到服务端IPv6地址, 翻墙固件不支持 IPv6 翻墙, 此种情况可能翻墙失败

解决方法是网络连接的属性里不要勾选 **Internet Protocol Version 6 (TCP/IPv6)**



🌐 全局不能翻墙, 首先ping 服务器的 ip 看看速度怎么样

```
ping 1.0.9.8
```

🐼 检查shadowsocks服务端启动时有没有带上 -u 参数

-u enable udprelay mode TPROXY is required in redir mode

本教程使用的, 也就是官方的shadowsocks-libev服务端是默认启动带上 -u 参数的。但有的朋友可能使用其他版本的服务端, 如Python版, 就不能保证服务端启动时默认就带 -u 参数

可以这样查询服务端是否启动, 及启动参数:

```
$ ps -aux | grep ss-server
#.../usr/bin/ss-server -c /etc/shadowsocks-libev/config.json -a root -u -f /var/run/shadowsocks-libev/shadowsocks-libev.pid
```

可见上面启动时已经带了 -u 参数

😁 登录OpenWrt路由器查询翻墙相关进程有没有启动

```
root@eastking:~# ps | grep ss-
#.../usr/bin/ss-redir -b 0.0.0.0 -c /etc/shadowsocks-libev/config.json -f /var/run/shadowsocks-libev/shadowsocks-libev.pid
#.../usr/bin/ss-tunnel -b 0.0.0.0 -c /etc/shadowsocks-libev/config.json -l 3210 -L 8.8.4.4:53 -u
```

```
root@eastking:~# ps | grep dnsmasq
#.../usr/sbin/dnsmasq -C /var/etc/dnsmasq.conf -k -x /var/run/dnsmasq/dnsmasq.pid
```

上面的查询显示, ss-redir ss-tunnel dnsmasq都已经正常启动

有时虽然ss-redir ss-tunnel dnsmasq等进程都在, 但已经失去响应了, 这就需要:

重启 shadowsocks, 登录路由器, 运行命令:

```
/etc/init.d/shadowsocks restart
```

restart 内部分 stop 和 start 两步执行, 实际测试发现, 少数时候 stop 并不能关闭 shadowsocks相关进程, 那么只能:

按路由器背后的电源重启OpenWrt路由器

翻墙不稳定, 有时能连上被墙网站, 有时连不上

shadowsocks-libev 加密翻墙的方式加大了墙的辨识难度, 但不是不可能被辨识。因此, 还是有可能受到干扰的。解决方法: 更换加密方式, 服务端 IP 地址或者服务端的端口

如果是轻度使用翻墙, 一般情况即使暂时翻墙不稳定, 问题也不大。如果重度使用翻墙, 可能被特别注意, 被干扰的机率会更大

登录路由器用dig查询被墙域名

本教程预编译的翻墙固件都安装了 bind-dig, 方便调试

注: 本教程默认的 tunnel 转发端口都是 3210

正常的结果类似如下:

```
root@eastking:~# dig @localhost -p 3210 google.com

; <<>> DiG 9.9.7-P3 <<>> @localhost -p 3210 google.com
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 38460
;; flags: qr rd ra; QUERY: 1, ANSWER: 11, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;google.com.                IN      A

;; ANSWER SECTION:
google.com.                299     IN      A       74.125.226.33
google.com.                299     IN      A       74.125.226.36
google.com.                299     IN      A       74.125.226.32
google.com.                299     IN      A       74.125.226.38
google.com.                299     IN      A       74.125.226.41
google.com.                299     IN      A       74.125.226.39
google.com.                299     IN      A       74.125.226.35
google.com.                299     IN      A       74.125.226.46
google.com.                299     IN      A       74.125.226.37
google.com.                299     IN      A       74.125.226.40
google.com.                299     IN      A       74.125.226.34

;; Query time: 290 msec
;; SERVER: 127.0.0.1#3210(127.0.0.1)
;; WHEN: Mon Dec 28 11:55:30 CST 2015
;; MSG SIZE rcvd: 215
```

仔细检查每一项翻墙配置

还是不能翻墙? 也有可能是某项配置有误, 仔细检查教程中讲到的每一项翻墙设置, 确保没有错误

有一次我给路由器新刷翻墙固件后, 总是不能翻墙。于是逐项检查, 发现了某项配置有误, 修正后就可用了

有几次, 我发现 /etc/dnsmasq.d/gfwlist.conf 文件成了空文件, 恢复后就正常了

相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2019-07-30

OpenWrt路由器为什么会翻墙失败或不稳定

- 🚗 有的网站能翻墙, 有的网站翻墙失败或者打开非常慢, 这是怎么回事
- 🌐 极少数网站比如 www.dropbox.com 打不开, 怎么办
- 🌐 全局不能翻墙, 首先ping 服务器的 ip 看看速度怎么样
- 🐼 检查shadowsocks服务端启动时有没有带上 -u参数
- 🐼 登录OpenWrt路由器查询翻墙相关进程有没有启动
- 🐼 重启 shadowsocks, 登录路由器, 运行命令:
- 🐼 按路由器背后的电源重启OpenWrt路由器
- 🐼 翻墙不稳定, 有时能连上被墙网站, 有时连不上
- 🐼 登录路由器用dig查询被墙域名
- 🐼 仔细检查每一项翻墙配置

Shadowsocks翻墙不同加密算法的区别



Shadowsocks翻墙不同加密方法，哪一种速度最快最好：

- 翻墙不稳定，有的能上，有的不能上，有时能上，有时不能上，可能是加密方式的特征被识别，从而被干扰，方法是更换加密方式
- rc4-md5解密速度虽然快，但是加密强度不够大，容易被干扰
- 无论哪一种加密方式，只要使用的人多了，就可能被重点研究，从而受到干扰
- 目前推荐使用 AEAD 加密方式
 - xchacha20-ietf-poly1305
 - chacha20-ietf-poly1305
 - aes-256-gcm
 - aes-192-gcm
 - aes-128-gcm

下列加密方法存在已知的弱点，不要使用：

```
bf-cfb
chacha20
salsa20
rc4-md5
```

下列加密方法已经不推荐了，可能会被探测到：

```
aes-128-ctr
aes-192-ctr
aes-256-ctr
aes-128-cfb
aes-192-cfb
aes-256-cfb
camellia-128-cfb
camellia-192-cfb
camellia-256-cfb
chacha20-ietf
```



什么是 AEAD 加密方法

缩写易忘是因为不知道原形，复杂之所以复杂是因为缺少细节的了解。世事莫不如此

AEAD 就是 Authenticated Encryption with Associated Data，使用关联数据进行身份验证加密，是一种同时具备保密性、完整性和可认证性的加密方法

201809 预编译 WNDR4300 翻墙固件已经支持目前最受推荐的 AEAD 加密实现之一：xchacha20-ietf-poly1305

那么什么是 xchacha20-ietf-poly1305 加密



推荐使用 xchacha20-ietf-poly1305 加密

xchacha20-ietf-poly1305 加密算法被 [libsodium](#) 官方推荐

```
which one should I use? XChaCha20-Poly1305-IETF is the safest choice.
```

我应该选择哪种加密算法？

XChaCha20-Poly1305-IETF是最安全的选择



怎样开启XChaCha20-Poly1305-IETF 加密算法

- 服务端，Ubuntu 17.10 或更新版本安装 shadowsocks-libev后，自动支持

我通常习惯把服Ubuntu更新到最新版，登录的欢迎页面显示是 18.04.1

再看下 shadowsocks-libev 的版本：

```
ss-server --help
```

```
shadowsocks-libev 3.1.3

-m <encrypt_method>      Encrypt method: rc4-md5,
aes-128-gcm, aes-192-gcm, aes-256-gcm,
aes-128-cfb, aes-192-cfb, aes-256-cfb,
aes-128-ctr, aes-192-ctr, aes-256-ctr,
camellia-128-cfb, camellia-192-cfb,
camellia-256-cfb, bf-cfb,
chacha20-ietf-poly1305,
xchacha20-ietf-poly1305,
salsa20, chacha20 and chacha20-ietf.
The default cipher is rc4-md5.
```

- 路由器是否支持 xchacha20-ietf-poly1305

路由器要支持xchacha20-ietf-poly1305加密, 需要满足二个条件:

```
- shadowsocks-libev 3.0+ (2017 年 2 月 1 日)
- libsodium 1.0.12+
```

实际上, 编译shadowsocks-libev for OpenWrt时会同时编译依赖库, 只要shadowsocks-libev 的版本满足条件就可以了

登录201809编译固件的 wndr4300 路由器查看

```
root@eastking:/etc# ss-redir -h
```

```
shadowsocks-libev 3.2.0

-m <encrypt_method>      Encrypt method: rc4-md5,
aes-128-gcm, aes-192-gcm, aes-256-gcm,
aes-128-cfb, aes-192-cfb, aes-256-cfb,
aes-128-ctr, aes-192-ctr, aes-256-ctr,
camellia-128-cfb, camellia-192-cfb,
camellia-256-cfb, bf-cfb,
chacha20-ietf-poly1305,
xchacha20-ietf-poly1305,
salsa20, chacha20 and chacha20-ietf.
The default cipher is rc4-md5.
```

考虑到 wndr4300 性能并不强悍, 可以使用chacha20-ietf-poly1305, 应该比 xchacha20-ietf-poly1305 节省一些资源

- Windows PC 客户端

[shadowsocks-windows](#) 自 4.0.9 版本(2018 年 3 月 14 日)起支持 xchacha20-ietf-poly1305 加密算法

- Android 客户端





[shadowsocks-android](#) 自 4.1.4 版本(2017 年 4 月 12 日)起支持 xchacha20-ietf-poly1305 加密算法

相关资源:

- <https://shadowsocks.org/en/spec/AEAD-Ciphers.html>
- <https://shadowsocks.org/en/spec/Stream-Ciphers.html>
- <https://github.com/shadowsocks/shadowsocks-org/issues/3>
- https://download.libsodium.org/doc/secret-key_cryptography/aead#tldr-which-one-should-i-use
- <https://zzz.buzz/zh/gfw/2018/03/18/shadowsocks-xchacha20-ietf-poly1305-compatibility-notes/>
- <https://zhuanlan.zhihu.com/p/28566058>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-10-20

Shadowsocks翻墙不同加密算法的区别

-  Shadowsocks翻墙不同加密方法, 哪一种速度最快最好:
-  什么是 AEAD 加密方法
-  推荐使用 xchacha20-ietf-poly1305 加密
-  怎样开启XChaCha20-Poly1305-IETF 加密算法

零起点DO VPS shadowsocks-libev 翻墙设置教程

Digital Ocean 的优点:

- 业界最有名的VPS服务商, 服务有保障
- 全SSD硬盘, 速度极快, 重启在10秒内
- 所有 VPS 具有独立 IP 地址
- 费用极低, \$5/月起, 作为 shadowsocks 足够了
- 管理后台Console Access可以直接运行所有linux命令, 可以不设置SSH
- 收费以小时计算, 不用了可以删除, 不会多收一分钱
- 官方专业人员发布大量零起点教程, 服务器管理菜鸟的福音
- 更换IP方便, 创建 snapshot, 再从 snapshot 新建 Droplet, 就能得到新的IP了

有时 VPS IP被屏蔽, 翻墙自然失败, 必须更换IP, Digital Ocean 更换 IP 就是几分钟的事情, 然后翻墙客户端更新一下 Server IP 就可以了

[立即点击这里注册DO](#)

创建翻墙用的虚拟服务器Droplet

注册DO并绑定支付方式后, 登录管理后台, 点击右上角的 **Create** 从下拉菜单中选择 **Droplets** :

- Choose an image 默认就是最新版的 Ubuntu x64, 挺好! shadowsocks-libev有的功能需要较新版的 Ubuntu 才能支持:

<https://github.com/softwaredownload/openwrt-fanqiang>

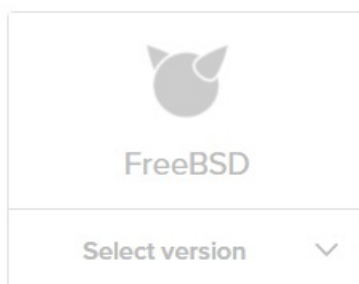
Create Droplets

Choose an image

Distributions

Container distributions

One-click



选择最新版本的 **Ubuntu** (默认)

- Choose a size 最便宜款配置足够强大了：

Choose a size

Standard Droplets


Balanced virtual machines with a healthy amount of memory tuned to host and scale applications like blogs, web applications, testing / staging environments, in-memory caching and databases.

MEMORY	vCPUs	SSD DISK	TRANSFER	PRICE
1 GB	1 vCPU	25 GB	1 TB	\$5/mo \$0.007/hr
2 GB	1 vCPU	50 GB	2 TB	\$10/mo \$0.015/hr

- Choose a datacenter region

数据中心选择, 8个城市可选, 一般选择 San Francisco 或 New York。据从国内有限ping ip测试, San Francisco 比 New York 快约60ms

Choose a datacenter region




New York

1

2


3



San Francisco

1


2



Amsterdam


2

3



Toronto

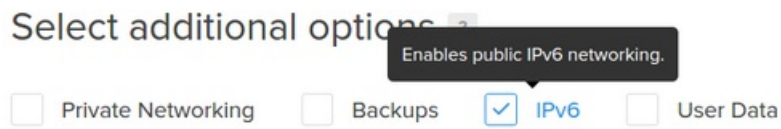
1



Bangalore

1

- Select additional options, 勾选IPv6:



- Choose a hostname, 只是助记, 比如改成ubuntu-shadowsocks
- Create 创建虚拟服务器

🐭 进入 Digital Ocean VPS 管理界面

在20秒内, VPS创建完毕后自动进入了 Droplets (VPS)列表页面, 点击VPS名字进入VPS管理界面:



🗨️ 如何重置DO VPS Root密码

注: 如果已经通过邮件收到root密码, 请跳到下一步

点击左边的 **Access** 再点击右边的 **Reset Root Password** 重置密码:

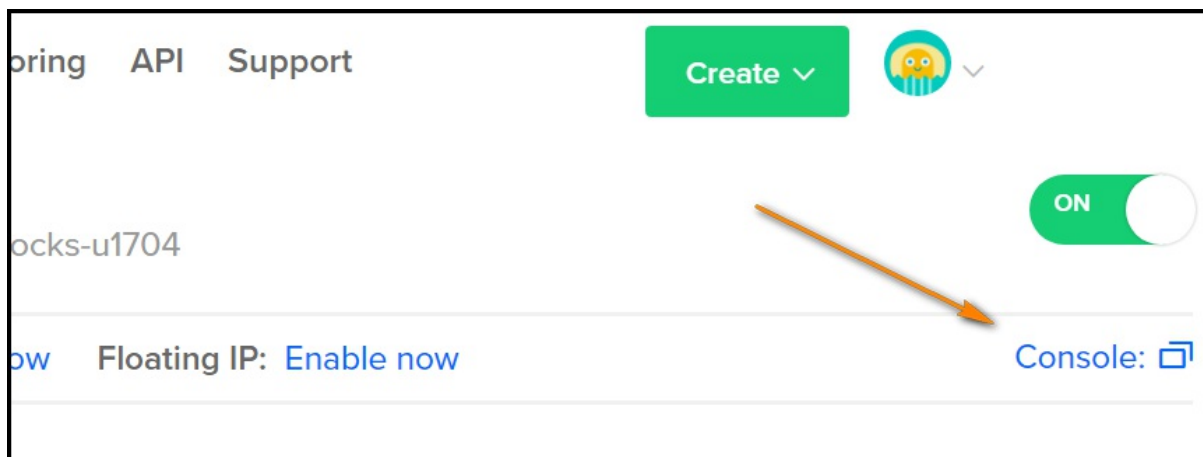
The screenshot shows the DigitalOcean control panel for a droplet. At the top, it displays '512 MB Memory / 20 GB Disk / NYC3 - Ubuntu shadowsocks'. Below this, network information is shown: 'ipv4: 45.55.85.128', 'ipv6: Enable now', and 'Private IP: Enable now'. A URL 'https://github.com/softwaredownload/openwrt-fanqiang' is listed. On the left sidebar, 'Access' is highlighted with an orange arrow. The main content area has two sections: 'Console access' with a description 'This will open up a console VNC connection keyboard directly to your virtual server.' and a blue 'Launch Console' button; and 'Reset root password' with a description 'This will shut down your Droplet and a new password will be generated for you.' and a question 'Do you wish to proceed?'. Below this is a grey 'Reset Root Password' button, which is also pointed to by an orange arrow.

重置密码完成后, 新的密码会发送到你的邮箱, 下面我们就用这个密码登录并直接通过网页 Console 控制台管理 VPS

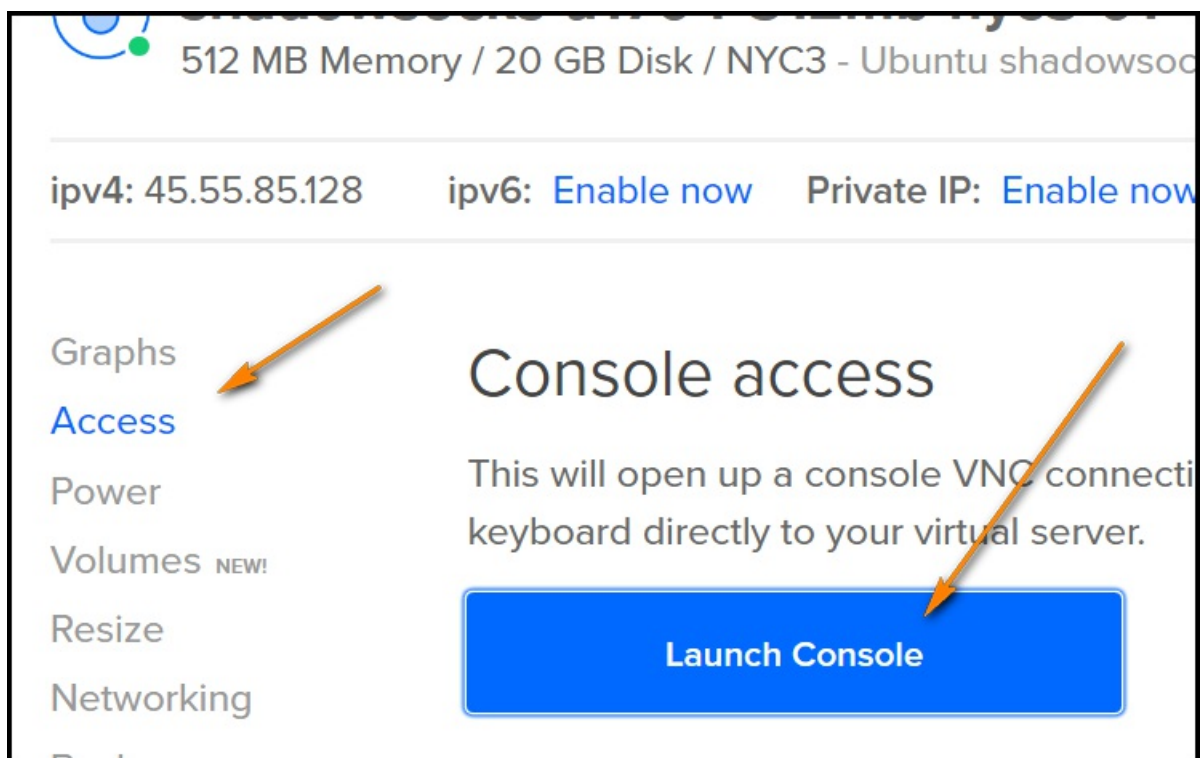
🕶️ 进入DO VPS命令行控制界面 **Console Access**

DO有个极为强大的功能, 可以可以直接在管理后台Console Access 运行Linux命令管理VPS, 相当于一个在线版的ssh

点击页面右上角的 **Console** 可以直接进入, 如下图:



打开 Console 的另一方法是先点击左边的 `Access` 然后点击 `Launch Console` 如下图：



如果等了一会儿命令行界面还没有出来，就按F5刷新页面直到打开

点击打开的命令行窗口以获得输入焦点

🤖 命令行设置新的Root密码

开启DO Console Access后，输入root并回车，然后重新设置密码

```
Ubuntu 18.04 ubuntu-shadowsocks tty1
ubuntu-shadowsocks login: root
Passwd: 输入root密码
You are required to change your password immediately (root enforced)
Changing password for root.
(Current) UNIX password: 输入root密码
Enter new UNIX password: 输入新的root密码
Retype UNIX password: 再次输入新的root密码
```

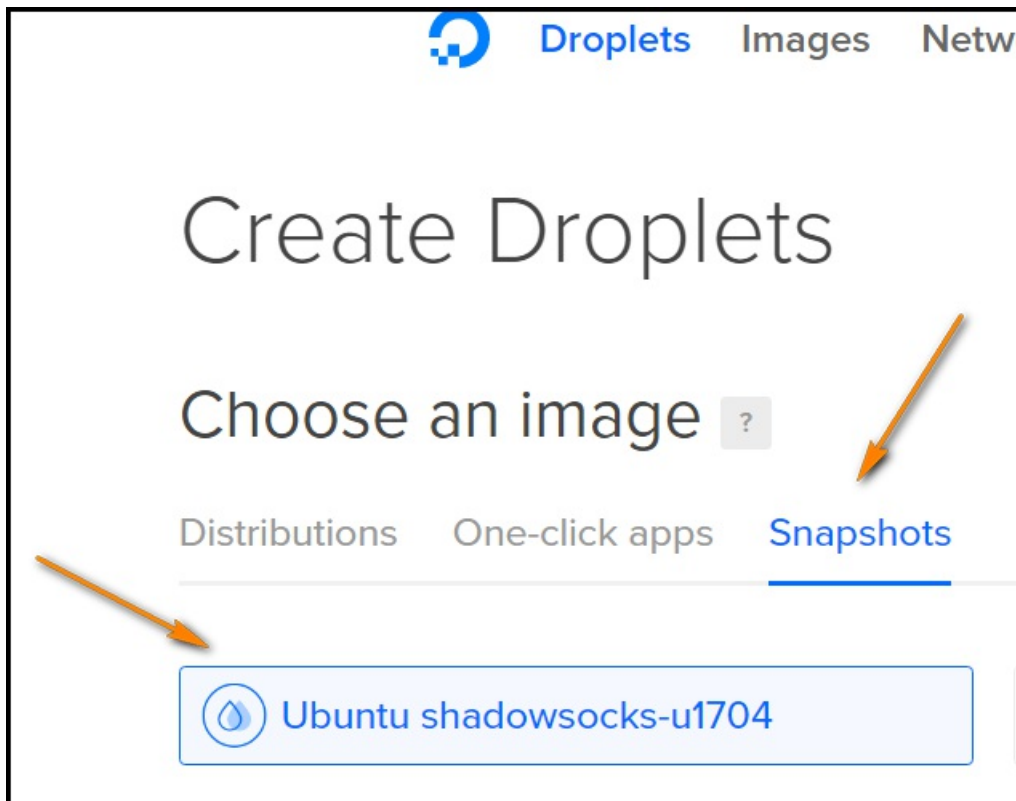
密码更新完成后更新一下系统：

```
root@ubuntu-shadowsocks:~# apt-get update
root@ubuntu-shadowsocks:~# apt-get dist-upgrade
```

可能会问你要不要更新一下grub, 直接回车就行了。(我选择的是升级到 `install the package maintainer's version`)

附录一：怎样快速更换DO翻墙VPS的IP(或者怎样使用最省钱)

- 照上面教程创建Droplet ubuntu-shadowsocks, 设置好shadowsocks-libev服务端, 其中server写 `0.0.0.0` 并测试通过
- Poweroff VPS, 也就是VPS关机, 这时还会产生VPS使用费用的, 因为IP, 空间等资源还是被你占用
- 创建Snapshot, 命名为shadowsocks, 并传送到你可能使用的各个区域。比如你原来是在San Francisco创建的, 可以传送到New York区
- 删除VPS: Destroy Droplet ubuntu-shadowsocks, 然后就不产生任何费用了。不怕麻烦, 每天都这样操作, 一个月可能只要几元钱就行了
- 下次要使用, 在Create Droplet的第一步, Choose an image, 选择Snapshots, shadowsocks, 其他和上面教程一样。见下图:



- 从snapshot创建Droplet完成, 页面显示了VPS的IP地址, shadowsocks客户端连接到这个IP地址就行了, 服务端不用更改任何设置

附录二：怎样不“登录”路由器更改OpenWrt shadowsocks-libev路由器的server IP

- 路由器设置密钥登录, 这样ssh登录就不用密码了
- 创建config配置文件, Ubuntu下是 `~/.ssh/config`, 增加如下内容:

```
Host router
  HostName 192.168.1.1
  User root
  Port 22
  IdentityFile /path/to/your/rsa
```

Windows下安装 [git for Windows](#), 选择使用OpenSSH, 编辑 `C:\Program Files\Git\etc\ssh\ssh_config`

然后同 Ubuntu 下一样可以用 `ssh router` 登录路由器, 再也不用手动输入密码登录路由器了

- `resetip.sh`:

```
#!/bin/bash

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date: 2016-01-20
```

```
ssh router <<'ENDSSH'

sed -ri "s/([0-9]{1,3}\.){3}[0-9]{1,3} -j/1.0.9.8 -j/" /usr/bin/ss-firewall-asia
sed -ri "s/([0-9]{1,3}\.){3}[0-9]{1,3}/1.0.9.8/" /etc/shadowsocks-libev/config.json

/etc/init.d/shadowsocks restart

ENDSSH
```

把resetip.sh中的 1.0.9.8 改成shadowsocks服务端的server IP, 然后运行 resetip.sh就可以了

Windows下安装 [git for Windows](#) 后, 用资源管理器打开 resetip.sh 所在目录, 右键, 选择 **Git Bash Here**, 然后 **./resetip.sh** 就可以执行 bash 脚本了

想要测试一下日本, 英国, 新加坡或美国的IP, so easy, 5分钟就行了

附录三: 江湖求急, 用DO Console 控制台从源码编译 shadowsocks-libev server

2016-01-19发现, shadowsocks.org 网页无法打开, 这给 apt-get install 方式安装shadowsocks-libev带来不便, 不过我们可以自己从源码编译, 很简单, 而且随时可以编译到最新的版本

Console Access 界面是无法粘贴命令的, 把下面命令逐行粘贴到浏览器地址栏, 抄着输入也是很快的, 输入第一行命令并回车后输入 y 安装所有相关包

```
root@ubuntu-shadowsocks:~# apt-get install build-essential autoconf libtool libssl-dev gawk debhelper dh-systemd init-system-helpers pkg-config git
root@ubuntu-shadowsocks:~# git clone https://github.com/shadowsocks/shadowsocks-libev.git
root@ubuntu-shadowsocks:~# cd shadowsocks-libev
root@ubuntu-shadowsocks:~# dpkg-buildpackage -us -uc -i
root@ubuntu-shadowsocks:~# cd ..
root@ubuntu-shadowsocks:~# sudo dpkg -i shadowsocks-libev*.deb
root@ubuntu-shadowsocks:~# ls /usr/bin/ss-*
root@ubuntu-shadowsocks:~# ss-local ss-manager ss-redir ss-server ss-tunnel
```

设置 shadowsocks-libev server, 见 [翻墙软件Shadowsocks-libev服务端设置](#)

至此, 我们已经开通了DO VPS,并且在网页界面就安装完成了 shadowsocks-libev, 下面是修改设置并重启shadowsocks-libev

```
root@ubuntu-shadowsocks:~# vi /etc/shadowsocks-libev/config.json
root@ubuntu-shadowsocks:~# service shadowsocks-libev restart
```

详细的设置教程在 [翻墙软件Shadowsocks-libev服务端设置](#)

再配置好客户端, 如果没有错误, 就可以成功翻墙了。所有以上过程2016-01-19亲测通过

一般情况下我们应该从仓库安装预编译shadowsocks-libev包:

```
sudo apt update
sudo apt install shadowsocks-libev
```

从仓库安装以后, 以后你 更新 Ubuntu 时, shadowsocks 也会得到更新(如果有新版)。如果你是从源码编译安装的 shadowsocks, 难道弄个定时器提醒自己十天半月从源码编译更新一次? 岂不烦人

这里从源码编译 shadowsocks-libev 只是演示 [DO Console](#) 控制台的强大之处。万一服务器 IP 被封, 或者其他原因登录不上 SSH, 我们可以用 DO Console 管理 VPS, Console 就是在线版的 SSH

VPS价格更便宜的也许会有, 但是服务稳定性, 技术积累, 各种资源, 小型 VPS 提供商是不能和 DO 这样业界领先的 VPS 提供商相比的

相关资源:

- <https://github.com/shadowsocks/shadowsocks-libev>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

[零起点DO VPS shadowsocks-libev 翻墙设置教程](#)

-  Digital Ocean 的优点:
-  创建翻墙用的虚拟服务器Droplet
-  进入 Digital Ocean VPS管理界面
-  如何重置DO VPS Root密码

-  进入DO VPS命令行控制界面 Console Access
-  命令行设置新的Root密码
-  附录一:怎样快速更换DO翻墙VPS的IP(或者怎样使用最省钱)
-  附录二:怎样不“登录”路由器更改OpenWrt shadowsocks-libev路由器的server IP
-  附录三:江湖求急, 用DO Console 控制台从源码编译 shadowsocks-libev server

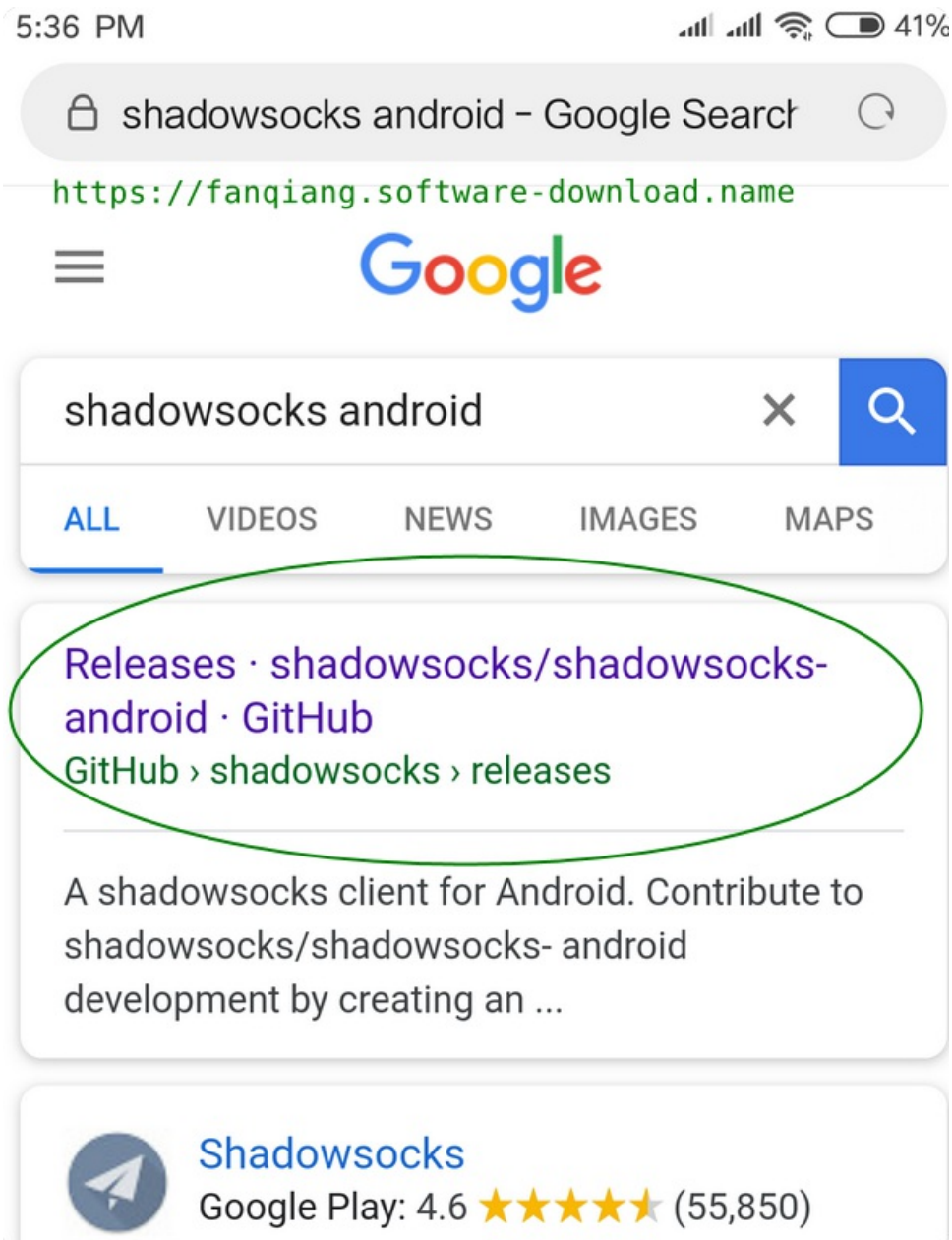
Android 安卓手机安装 shadowsocks 影梭翻墙、科学上网教程

本教程可以用于小米、华为、三星、VIVO、OPPO等安卓手机安装 shadowsocks 翻墙

下载 shadowsocks-Android 安卓版翻墙软件

如果手机已经连上了已经翻墙的路由器, 那么可以这样操作:

- 打开浏览器, 地址栏输入 <https://www.google.com>
- 搜索 `android shadowsocks`



- 如上图, 第一个搜索结果点击进去就是了

如果手机没有连上翻墙网络, 那么可以直接从开源项目地址 github 下载。点击下面链接:

<https://github.com/shadowsocks/shadowsocks-android/releases>

- 进入了Android shadowsocks 下载页面, 如下图这样:

Releases · shadowsocks/shadowsocks-r

<https://fanqiang.software-download.name>

shadowsocks / shadowsocks-r



Code

Issues 9

Pull requests 2

Projects 0

Releases

Tags

v4.6.1

Pre-release

v4.6.1 8aabc73



madeye released this on Jul 4 · 153 commits to master since v4.6.0 release

> Assets 6

Add Android P support.

- 点击 Assets 展开下载文件列表, 如下图:

5:36 PM 41%

Code Issues 9 Pull requests 2 Projects 0
<https://fanqiang.software-download.name>

Releases Tags

v4.6.1

Pre-release

v4.6.1
madeye re
master since this

1. 点击 Assets

Assets 6

shadowsocks--universal-4.6.1.apk	9.09 MB
shadowsocks-arm64-v8a-4.6.1.apk	5.26 MB
shadowsocks-armeabi-v7a-4.6.1.apk	5.16 MB
shadowsocks-x86-4.6.1.apk	5.3 MB

2. 选择apk下载

点击 shadowsocks-arm64... 或者 shadowsocks--universal... 开始下载。手机安装第三方软件的方法可以搜索手机品牌的软件安装教程

Android 安卓手机设置 shadowsocks 翻墙配置文件

- 启动 shadowsocks后, 点击右上角的加号创建新的科学上网配置文件, 如下图



- Android shadowsocks 翻墙参数设置

5:28 PM42%

×

Profile config

🗑️

✓

Profile Name 配置名称

fanqiang.software-download.name

Server Settings

Server 服务端IP

1.0.9.8

Remote Port 远程端口

1098

Password 密码

.....

Encrypt Method 加密方式

CHACHA20-IETF-POLY1305

🔒

https://fanqiang.software-download.name

Feature Settings

Route 路由

Bypass LAN & mainland China

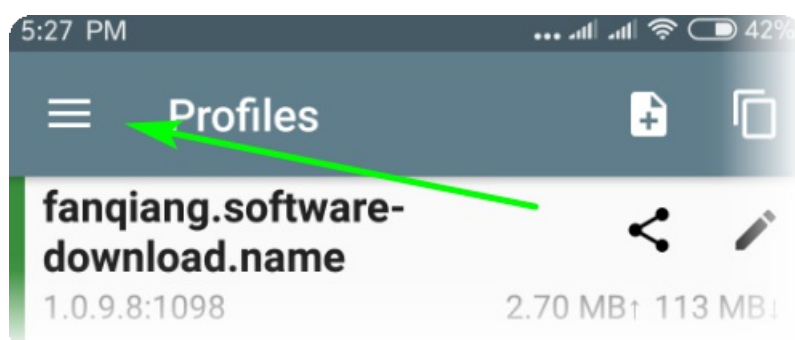
- Profile Name 配置名称, 只是助记。这里写了 fanqiang.software-download.name
- Server 服务器IP地址, 1.0.9.8 改成你自己的
- Remote Port 远程端口, 1098 改成实际端口
- Password 密码
- Encrypt Method 加密方法, 推荐 CHACHA20-IETF-POLY1305

- Route 路由，一般选 Bypass LAN and mainland China 绕过局域网及中国大陆地址
- 点右上角的对号保存翻墙设置，如果以后再次编辑也是点击这里。如下图

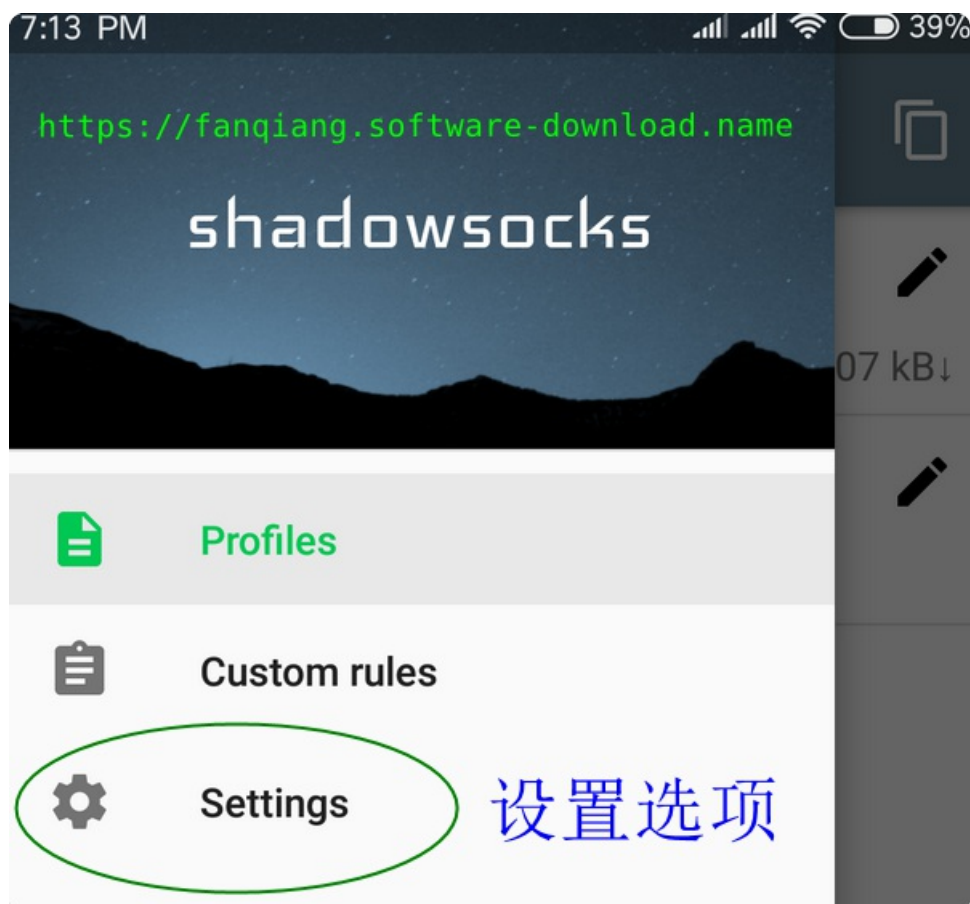


🚗 Android 安卓手机 shadowsocks 设置选项

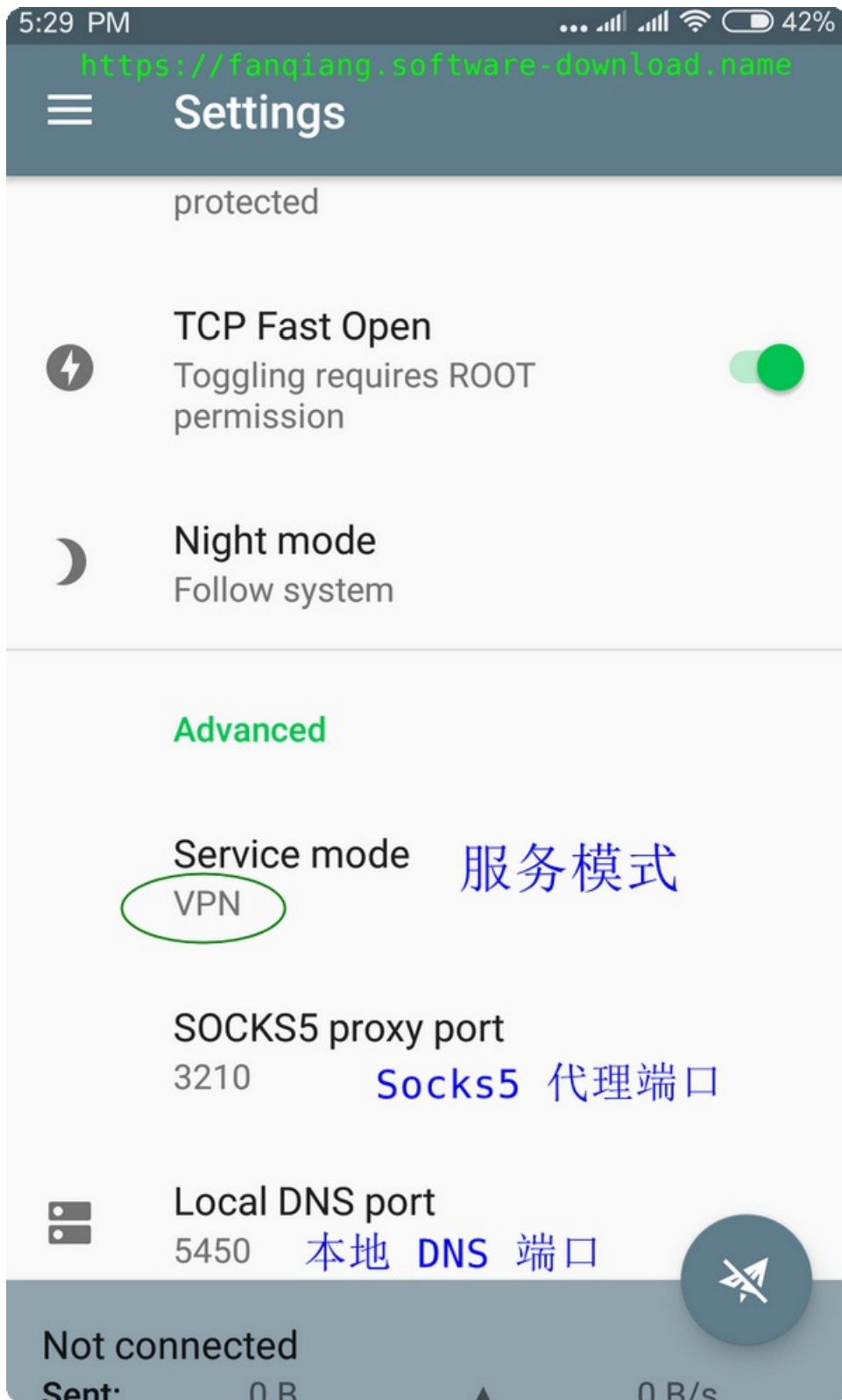
- 点击左上角的三个横杠，进入高级设置界面，如下图：



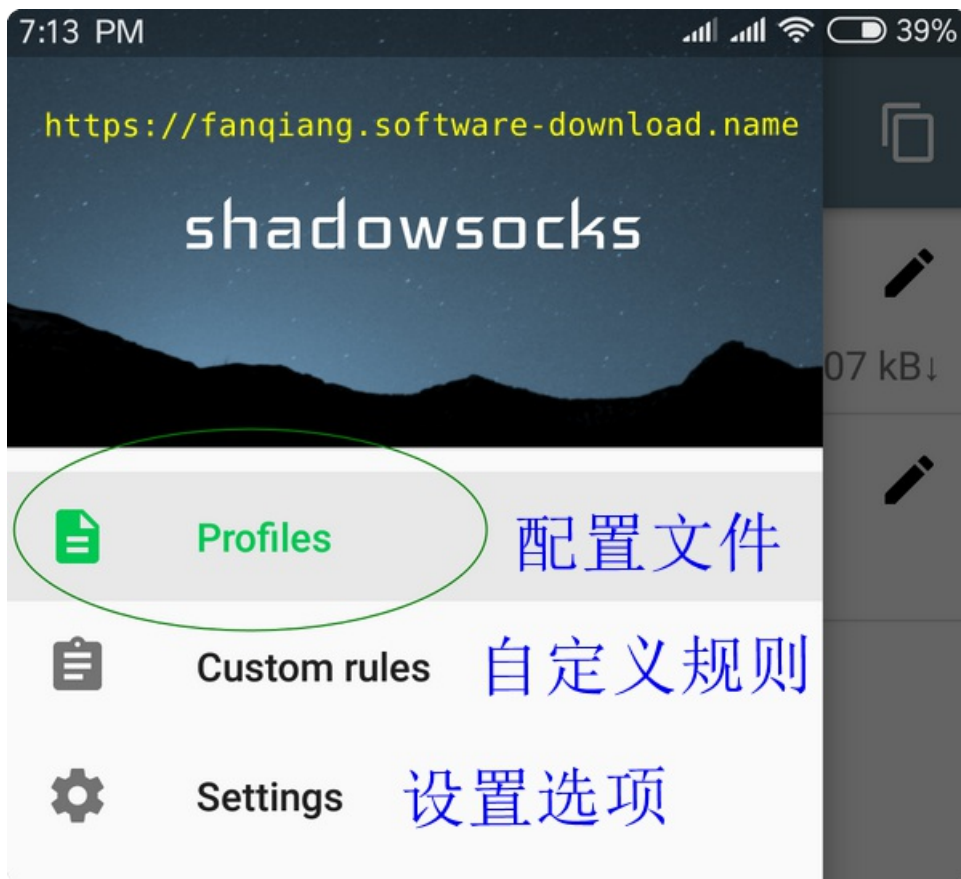
- 选择 Settings 设置选项，如下图：



- Service mode 服务模式, 选 VPN, 如下图:



- 如果 shadowsocks 服务端开启了 TCP fast open, 则可以滑动开启 TCP Fast Open。图中已经开启了, 翻墙速度会有提升
- 再次点击左上角三横杠, 然后点击 Profiles 配置文件 回到配置文件界面, 如下图:



😬 Android shadowsocks 安卓手机影梭科学上网测试

- 点击配置文件名称 `fanqiang.software-download.name` 以选中配置文件，选中后左边有绿色的竖条

你可能会问，我就一个配置文件，为什么还要选中配置文件才能使用？这是考虑到有的人可能拥有多个服务端，或者测试多个配置文件，于是左边有绿色的竖条表示选中的配置，如果是灰色竖条就表示没有选中，很直观

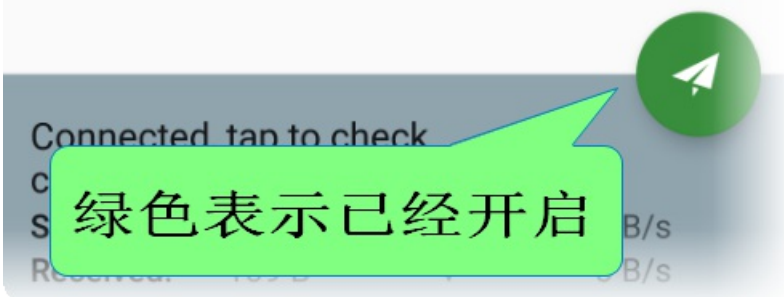
如果要删除一个不需要的配置文件，向左滑动就可以了

- 右下角有个圆形图标，中间有个梭子，灰色表示没有开启翻墙，点击它，图标会变成绿色，表示 shadowsocks 已经在连接服务端



如下图, 绿色图标表示 shadowsocks 已经在工作了

`https://fanqiang.software-download.name`



- 测试安卓手机翻墙有没有成功

如果手机已经连接了翻墙路由器, 先关闭WIFI 连接, 改用流量上网

打开浏览器, 导航到:

<https://m.youtube.com>

如下图, 用手机流量科学上网看 youtube, 速度挺不错

7:45 PM

4G 36%

YouTube



Save the Jeep

HDBroadcaster .com 44,654,671 views

35K

21K



<https://fanqiang.software-download.name>

Up next

Autoplay ☒



Save the Girls

HDBroadcaster .com
319K views

如果打不开 youtube, 就要仔细检查 各项设置是否正确

最后提醒一下, 科学上网结束时, 要点击 shadowsocks-Android 右下角圆形图标停止翻墙, 否则手机电量会哗哗往下掉

相关资源:

- <https://github.com/shadowsocks/shadowsocks-android/releases>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/ebook/06.01.md>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name>

2018-12-07

Android 安卓手机安装 shadowsocks 影梭翻墙、科学上网教程

-  下载 shadowsocks-Android 安卓版翻墙软件
-  Android 安卓手机设置 shadowsocks 翻墙配置文件
-  Android 安卓手机 shadowsocks 设置选项
-  Android shadowsocks 安卓手机影梭科学上网测试

OpenWrt + Git Bash for Windows 快速切换翻墙模式: 全局翻墙或局部翻墙

本项目 `/openwrt-fanqiang/bin` 下有三个文件, 用来切换不同的翻墙模式, 分别是:

- 翻墙时忽略亚洲IP: `ss-firewall-asia`
- 全局翻墙模式, 所有流量加密: `ss-firewall-global`
- 翻墙时忽略中国IP: `ss-firewall-china`

为什么要作这样的细分?

- 有些外网, 如果不是全局翻墙, 可能打不开
- 如果翻墙时忽略中国IP, 因中国区IP列表较长, 对有些路由器压力较大, 因此默认翻墙时忽略亚洲IP

如果你的路由器里面没有这几个文件, 请先把它们复制到路由器里

怎样手动切换翻墙模式

本项目 `/openwrt/default/etc/init.d/shadowsocks` 文件里有如下代码:

```
/usr/bin/ss-firewall-asia
#/usr/bin/ss-firewall-global
#/usr/bin/ss-firewall-china
```

切换方法是命令行登录路由器, 修改这四行代码, 把不需要的注释掉(以#开始), 把需要的行启用(去掉开始处#)

如果是一次性修改, 命令行切换翻墙模式也不麻烦

更常用的场景是, 平时设置翻墙时忽略中国或亚洲IP, 浏览外网时, 某些外网可能打不开, 这时需要切换到全局翻墙模式, 如果每次都命令行登录路由器手动切换, 就有点费时了

有没有更加简单的方便, 特别是在 Windows 下?

OpenWrt 路由器设置 ssh 免密码登录原理

网上教程很多, 可以用 `openwrt 免密码` 搜索教程了解详细原理, 这里略过

安装 Git for Windows

安装并设置 Git for Windows后, 我们就有了一个类似 Linux 下的 bash 环境, 做到不同系统操作习惯类似, 带来了很大的便利

- 下载地址: <https://git-scm.com/download/win>
- Select Components

选择组件 步骤时, 确认选中以几项(默认已经选中)

- Windows Explorer integration

和 Windows 资源管理器整合, 安装完成后, 在文件夹右击, 就可以 `Git Bash here` 打开当前目录下的 bash, 十分方便

- Associate .sh files to be run with Bash

.sh 文件由 Bash 执行, 和Linux 下一样, 双击 .sh 文件可以运行了

- Choosing the default editor used by Git

选择默认编辑器, 默认是Vim, 联准了:)

- Adjusting your PATH environment

调整 PATH 环境变量, 默认是选中第一项 `Use Git from Git Bash onley`, 如果只是从 Git Bash使用Git, 那么选中这项就可以了

选中 `Use Git from the Windows Command Prompt` 好处是可以让安装程序把 `git.exe` 的目录加入系统 Path 环境变量, 于是其他软件也可以从命令行调用 git 了

- Choosing the SSH executable

默认是 Use OPenSSH，很好，我们正需要和 Linux 下 ssh 操作习惯一致

最简单安装方法，全部安装默认。如果有需要修改的地方，可以重新再安装一次

配置 ssh config，实现自动登录路由器

64位系统，安装64位git，默认安装目录是：

C:\Program Files\Git

ssh 系统config文件是：

```
C:\Program Files\Git\etc\ssh\ssh_config
```

不建议把自定义设置写在这个文件里，以免重装Git后被覆盖。下面把自定义设置放在用户设置里

按 Windows 键，输入 git bash 回车，默认进入的是 \$HOME 目录

以下操作是 Linux 下一样的

```
# 列出当前目录，也就是 C:\Users\your_name 下的内容
$ ls

$ mkdir .ssh
$ cd .ssh
$ ls
$ touch config
$ vi config
```

输入下面内容：

```
Host router
  HostName 192.168.1.1
  User root
  Port 22
  IdentityFile /path/to/rsa
```

如果配置正确，运行下面命令就可以自动登录路由器(router)了

```
# 自动登录路由器
$ ssh router
```

如果你电脑里的所有重要文件都保存在云盘，那么可以创建链接文件，这时 \$HOME/.ssh/config 只是个链接，实际文件在云盘里，删除链接文件并不会删除实际文件

- 按 Windws + X
- Command Prompt(Admin) 控制台(管理员)

执行如下命令：

```
C:\WINDOWS\system32> cd %homepath%
C:\Users\name> cd .ssh
C:\Users\name\.ssh> del config
C:\Users\name\.ssh> mklink config C:\cloud_app\ssh\config
```

路由器一键切换四种翻墙模式

配置好免密码登录路由器后，大功已经成就了一半

创建一个 test.sh，内容如下：

```
#!/bin/sh

ssh router <<'ENDSSH'

# Arbitrary commands here execute on router
```

ENDSSH

前面我们在安装 Git for Windows 的时候, 已经选中 .sh 文件由 Bash 执行, 这时双击 test.sh, 就会自动登录路由器并执行中间的命令

比如, 我们创建 ss-global.sh, 双击, 就能自动切换到路由器全局翻墙模式:

ss-global.sh:

```
#!/bin/sh

ssh router <<'ENDSSH'

sed -i -e 's@^\(s*\)(/.\+ss-firewall\)\@1#\2@g' -e 's@^\(s*\)#\(/.\+ss-firewall-global$\)\@1\2@' /etc/init.d/shadowsocks
/etc/init.d/shadowsocks restart

ENDSSH
```

2018-10 起, 本项目 `openwrt-fanqiang/bin` 下新增几个文件用来切换路由器的翻墙模式:

- ss-asia 翻墙时忽略亚洲IP
- ss-global 切换到全局翻墙模式
- ss-china 翻墙时忽略中国IP

给它们加上 .sh 后缀并放在桌面, 就可以一键切换翻墙模式了

🔗 git bash 快速切换四种翻墙模式

如果上面几个文件不是放在桌面，就要先进入特定目录才能执行命令，这种情况下有没有更加简便的办法呢

办法有很多，我们可以把这四个文件的目录加入到 git bash 的 \$PATH 环境变量中，然后在 bash 中输入文件名就可以自动执行命令了

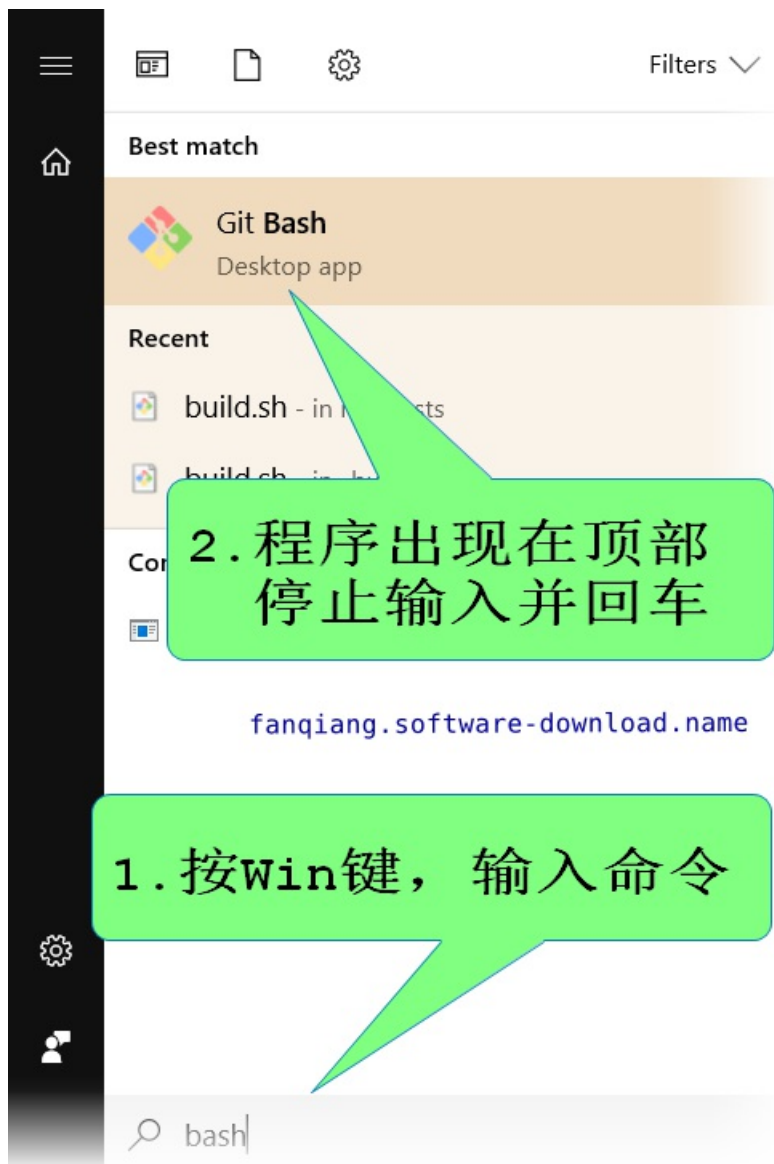
按 Windos 键, 输入 `git bash` 回车 调出 bash

假设你把本项目 <https://github.com/softwaredownload/openwrt-fanqiang> clone 到了 C 盘根目录, 在 Git Bash 里执行如下命令:

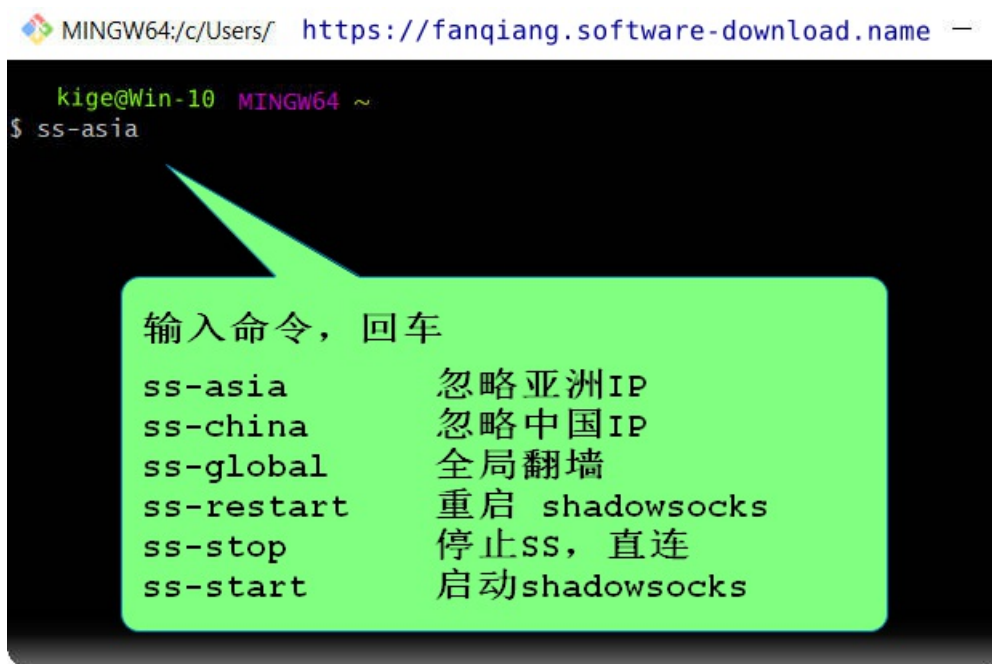
```
$ vi ~/.bashrc

# add line to it
PATH = "$PATH:/c/openwrt-fanqiang/bin"
```

也就是在 bash 环境变量 PATH 后面加上特定目录, 设置好后关闭 Git Bash 再调出以使修改生效



上图，Windows 10 下快速调出 Git Bash for Windows



上图, 在 Git Bash 命令提示符里输入命令, 回车执行。一般的 Linux 脚本, 都可以这样在Windows下执行

切换翻墙模式应用场景:

- 浏览外网, 某网打不开或打开很慢
 - 按Windows键, 输入关键词, 回车, 调出 Git Bash
 - 输入 `ss-global` 回车开始全局翻墙
 - 浏览外网结束, 调出 Git Bash
 - 输入 `ss-asia` 回车, 翻墙忽略亚洲IP

相关资源:

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/bin>
- <https://git-scm.com/download/win>
- <https://stackoverflow.com/questions/10681101/git-bash-doesnt-see-my-path>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

OpenWrt + Git Bash for Windows 快速切换翻墙模式: 全局翻墙或局部翻墙

-  怎样手动切换翻墙模式
-  OpenWrt 路由器设置 ssh 免密码登录原理
-  安装 Git for Windows
-  配置 ssh config, 实现自动登录路由器
-  路由器一键切换四种翻墙模式
-  git bash 快速切换四种翻墙模式
-  切换翻墙模式应用场景:

OpenWrt路由器编译翻墙固件教程

实践前面的教程，翻墙已经不是问题，白脸也很happy。在这一章中，我们要定制自己OpenWrt固件，刷上定制的固件，不用任何设置就自动翻墙并自动更新规则

最简单的路由器刷OpenWrt翻墙方案：

- <https://github.com/softwaredownload/openwrt-fanqiang>

在线阅读OpenWrt路由器翻墙、科学上网器教程：

- <https://fanqiang.software-download.name>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-10-22

编译shadowsocks-libev for OpenWrt ipk安装包

不同OpenWrt版本下编译的shadowsocks-libev ipk一般是不能通用的。比如现在路由器用的是18.06.1版的OpenWrt, 如果使用OpenWrt Chaos Calmer 15.05 下编译的shadowsocks-libev, 可能安装后根本不能启动

如果你懒得自己编译, 可以到下面地址下载:

<https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>

以下 不要使用root用户来操作

用OpenWrt SDK 编译 ipk(2018-09-22更新)

SDK是一个可重定位, 预编译的OpenWrt toolchain(工具链), 适用于为特定目标交叉编译单个用户空间包, 无需从头开始编译整个系统

使用SDK的原因是:

- 为特定版本编译自定义软件, 同时确保二进制和功能兼容性
- 编译某些软件包的较新版本
- 使用自定义修补程序或不同功能重新编译现有包

在2016年及以前的文章中, 我们没有使用 SDK, 自己重新编译一遍 toolchain 要花费较长时间, 现在用预编译的 SDK 可以大大节省编译 ipk 的时间

获取SDK:

您可以下载已编译的SDK, 也可以使用“make menuconfig”命令自行编译

如果编译 ipk 所使用的 OpenWrt 版本和 固件的 OpenWrt 版本不同, 那么可能会有兼容性问题。为了确保兼容, 我们在同一个页面下载 imagebuilder 和 SDK

今天是2018年9月22日, OpenWrt 适用于 ar71xx/nand 最新稳定版 18.06.1 在这个页面下载(适合于WNDR4300路由器):

<http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/nand/>

注意, 此处只是 Ubuntu 64bit 上示范编译 shadowsocks-openwrt-libev 的过程, 不同的路由器的下载目录可能不同

先决条件:

请参阅 [OpenWrt Buildroot](#) 页面以安装所需的软件以在SDK上构建软件包

注意: 在某些主机上, 需要安装ccache

Ubuntu 编译shadowsocks-libev for OpenWrt 步骤:

- 安装 ccache

```
sudo apt install ccache
```

- 下载 OpenWrt-SDK

```
cd ~/Downloads
wget http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/nand/openwrt-sdk-18.06.1-ar71xx-nand-gcc-7.3.0_musl.Linux-x86_64.tar.xz

tar -xf openwrt-sdk-18.06.1-ar71xx-nand-gcc-7.3.0_musl.Linux-x86_64.tar.xz
mv openwrt-sdk-18.06.1-ar71xx-nand-gcc-7.3.0_musl.Linux-x86_64 openwrt-sdk-nand
cd openwrt-sdk-nand
```

- 添加 feeds

```
git clone https://github.com/shadowsocks/openwrt-feeds.git package/feeds
```

- 获取 shadowsocks-libev Makefile

```
git clone https://github.com/shadowsocks/openwrt-shadowsocks.git package/shadowsocks-libev
```

- [最大化控制台](#), 否则可能有错误

```
include/toplevel.mk:136: recipe for target 'menuconfig' failed
```


- 选择要编译的包 Network -> shadowsocks-libev

```
make menuconfig
```

选择 `Network` --> 回车进入，光标移动到 `shadowsocks-libev-server` 按 `n` 取消选择，`Exit` 退出

退出时会询问：

```
Do you wish to save your new configuration?
```

Yes上回车

- 开始编译

```
make package/shadowsocks-libev/compile V=99
```

输出文件：

```
$:~/Downloads/openwrt-sdk-nand/bin$ tree .
.
├── packages
│   └── mips_24kc
│       ├── base
│       │   ├── libmbdtdls_2.12.0-2_mips_24kc.ipk
│       │   └── shadowsocks-libev_3.2.0-1_mips_24kc.ipk
│       └── packages
│           ├── libcures_1.14.0-1_mips_24kc.ipk
│           ├── libev_4.24-1_mips_24kc.ipk
│           ├── libpcrc_8.42-1_mips_24kc.ipk
│           └── libsodium_1.0.16-1_mips_24kc.ipk
└── targets
    └── ar71xx
        └── nand
            └── packages
                ├── libatomic_7.3.0-1_mips_24kc.ipk
                ├── libc_1.1.19-1_mips_24kc.ipk
                ├── libgcc_7.3.0-1_mips_24kc.ipk
                ├── libpthread_1.1.19-1_mips_24kc.ipk
                ├── librt_1.1.19-1_mips_24kc.ipk
                └── libstdc++_7.3.0-1_mips_24kc.ipk
```

packages\mips_24kc\下 shadowsocks-libev是主文件，除libpcrc以外的四个文件是必须依赖，编译翻墙固件时，把相关 ipk 复制到 imagebuilder/packages 目录下就可以了

相关资源：

- <https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>
- <https://github.com/shadowsocks/openwrt-shadowsocks>
- <https://openwrt.org/docs/guide-developer/obtain.firmware.sdk>
- <https://openwrt.org/docs/guide-developer/build-system/install-buildsystem>

以下是截止2016年4月的内容，可与上面用SDK编译的方法对照：-----

下面是在Ubuntu 64bit下编译shadowsocks-libev for OpenWrt ipk安装包的步骤：

🏠 安装依赖库，不同的操作系统版本可能要作相应调整

```
sudo apt-get install build-essential subversion libncurses5-dev zlib1g-dev gawk gcc-multilib flex git-core gettext
```

🐼 下载OpenWrt源代码

```
cd ~/Downloads
git clone git://git.openwrt.org/openwrt.git
```

🏈 下载shadowsocks-libev源码

```
cd ~/Downloads/openwrt
pushd package
git clone https://github.com/shadowsocks/shadowsocks-libev.git
popd
```

或者：

```
cd ~/Downloads/openwrt/package
git clone https://github.com/shadowsocks/shadowsocks-libev.git

编译 DIR505固件2015-12版时用的源码版本是:Date: Tue Dec 22 21:42:40 2015
```

更新Feeds

使package在make menuconfig中可用，而不是真正安装或编译，并按照自己的路由型号设定target，否则默认target下编译好的工具链在重新设定target后无效

```
cd ~/Downloads/openwrt
./scripts/feeds update -a
./scripts/feeds install -a
# run make menuconfig and set target;
# Choose your own Target System -> SubTarget -> Target Profile
make menuconfig
make defconfig
```

先编译要用到的工具和库

```
make prereq && make tools/install && make toolchain/install
```

等待时间较长，可以先和大妈一起去跳个广场舞，制造更多噪音恶心一下别人：)

make menuconfig配置选项

```
# 运行命令
make menuconfig
```

有三个选项：

- y: 编译进固件
- m: 编译出安装包，但不打包进固件
- n: 排除

输入命令 `make menuconfig` 进入配置程序后：

- Target System:
 - Atheros AR7xxx/AR9XXX (Default value, 不同的路由器，可能选择不同)

适合：WNDR4300, DIR505A1, TLWR2543

- Subtarget:
 - Generic device with NAND flash

适合：WNDR4300

- Generic

适合：DIR505A1

- Target Profile: (因我们只是编译包，这步可以不选)
- Network, 选择shadowsocks-libev 和 shadowsocks-libev-polarssl, 按m设置为编译独立ipk安装包
- Save && Exit

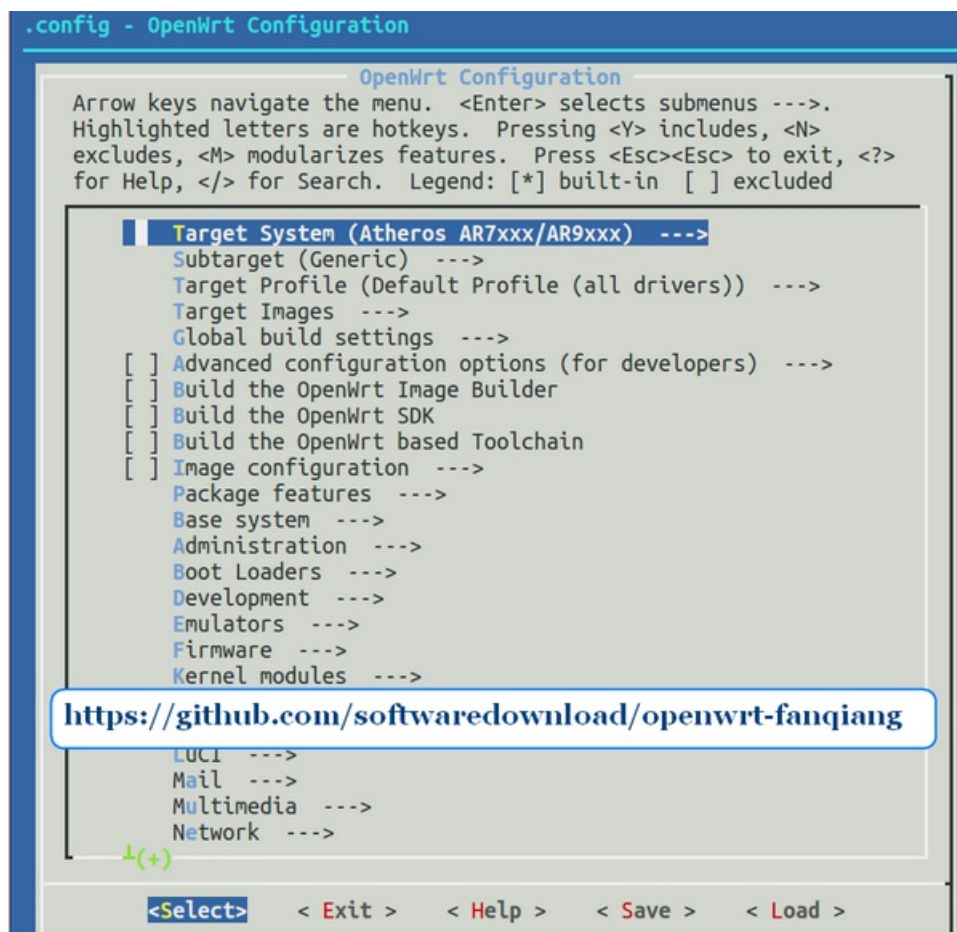


图 make menuconfig

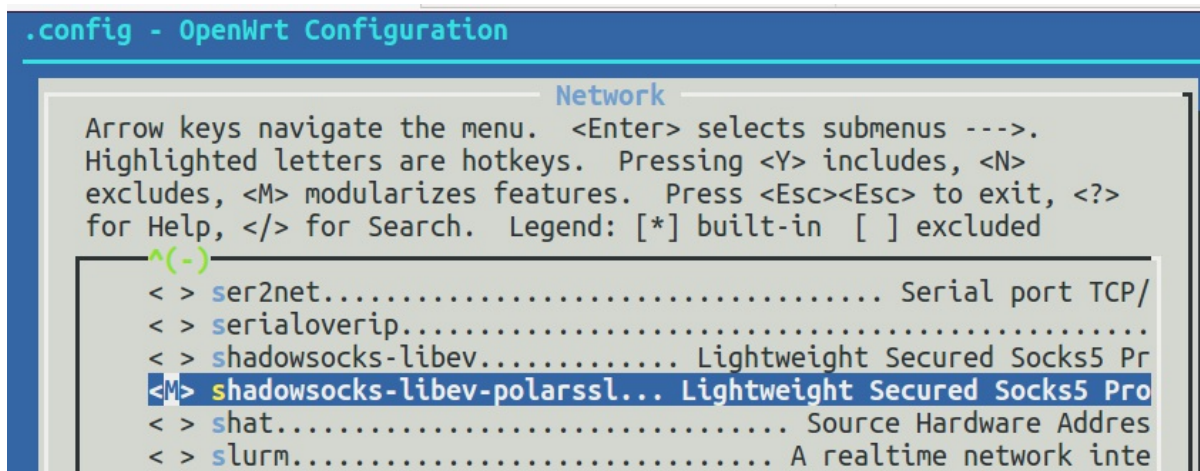


图 选择shadowsocks-libev-polarssl

编译shadowsocks-libev for OpenWrt

```
make V=99 package/shadowsocks-libev/openwrt/compile
```

查看编译出的shadowsocks-libev和shadowsocks-libev-polarssl文件

```
cd ~/Downloads/openwrt/bin/ar71xx/packages/base/
tree
├─ libc_1.1.11-1_ar71xx.ipk
├─ libgcc_5.2.0-1_ar71xx.ipk
└─ libopenssl_1.0.2e-1_ar71xx.ipk
```

```

├─ libpolarssl_1.3.15-1_ar71xx.ipk
├─ libpthread_1.1.11-1_ar71xx.ipk
├─ shadowsocks-libev_2.4.3_ar71xx.ipk
├─ shadowsocks-libev-polarssl_2.4.3_ar71xx.ipk
└─ zlib_1.2.8-1_ar71xx.ipk

# 如果用来编译翻墙固件, 把shadowsocks-libev复制到Image Builder目录下:
# for DIR505A1:
cp shadowsocks* ~/Downloads/openwrt-imagebuilder-ar71xx-generic.Linux-x86_64/packages/base
# for WNDR4300
cp shadowsocks* ~/Downloads/openwrt-imagebuilder/packages/base

```

把文件scp复制到OpenWrt路由器/tmp, 就可以 `opkg install shadowsocks-libev_2.4.3_ar71xx.ipk` 安装了

相关资源:

- <https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>
- <https://openwrt.org/zh-cn/doc/howto/buildroot.exigence>
- <https://openwrt.org/zh-cn/doc/howto/build>
- <https://github.com/shadowsocks/shadowsocks-libev>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-10-22
编译shadowsocks-libev for OpenWrt ipk安装包

-  用OpenWrt SDK 编译 ipk(2018-09-22更新)
-  安装依赖库, 不同的操作系统版本可能要作相应调整
-  下载OpenWrt源代码
-  下载shadowsocks-libev源码
-  更新Feeds
-  先编译要用到的工具和库
-  make menuconfig配置选项
-  编译shadowsocks-libev for OpenWrt
-  查看编译出的shadowsocks-libev和shadowsocks-libev-polarssl文件

下载和设置OpenWrt路由器翻墙配置文件

自己手工收集编辑翻墙所用到的配置文件是件比较累的事情。最快的方法是 git clone 本项目，修改其中某些选项

下载翻墙配置文件

```
cd ~/Downloads
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

默认配置文件目录: openwrt-fanqiang/openwrt/default

针对特定路由器的配置文件目录, 以路由器型号为目录名, 如 openwrt-fanqiang/openwrt/wndr4300


复制配置文件, 以wndr4300路由器为例:

- 本地建立配置文件目录, 如 ~/Downloads/openwrt-wndr4300
- 复制默认配置文件到 ~/Downloads/openwrt-wndr4300

```
mkdir ~/Downloads/openwrt-wndr4300

# Linux下复制默认配置文件
cp -R ~/Downloads/openwrt-fanqiang/openwrt/default/* ~/Downloads/openwrt-wndr4300/

# 复制WDR4300路由器的特定配置文件, 同名文件就覆盖
cp -R ~/Downloads/openwrt-fanqiang/openwrt/wndr4300/* ~/Downloads/openwrt-wndr4300/
```

 修改配置文件, 编译后就直接可以用了。否则刷上固件后登录路由器再修改。主要修改如下文件:

```
~/Downloads/openwrt-wndr4300/etc/shadowsocks-libev/config.json
~/Downloads/openwrt-wndr4300/usr/bin/ss-firewall-asia
~/Downloads/openwrt-wndr4300/etc/uci-defaults/defaults
```

- shadowsocks.json 中 server必须改成你的服务器实际IP
- defaults 中wan-username 和 wan-password必改
- ss-firewall 中 1.0.9.8必须改成你的服务器实际IP
- 编译自定义固件时, 设置FILES=~/Downloads/openwrt-wndr4300

自定义配置文件用途说明

定制固件的前提是你需要有一台服务器运行shadowsocks服务端ss-server

- etc/dnsmasq.conf 设置dnsmasq配置文件目录
- etc/shadow 登录路由器的密码, 默认是fanqiang
- etc/uci-defaults/defaults 默认上网设置及时区等设置

关于 /etc/uci-defaults目录

uci-defaults目录下的文件会在路由器第一次启动时由/etc/init.d/boot执行,如果在文件末尾加上exit 0, 则执行就会删除此文件, 否则执行成功则删除, 不成功则在下次启动时继续执行直到成功

我们在这个目录下创建一个defaults文件, 在这个文件中设置上网参数, 时区等

To set some system defaults the first time the device boots, create a script in the folder

All scripts in that folder are automatically executed by /etc/init.d/boot and if they exited with code 0 deleted afterwards (scripts that did not exit with code 0 are not deleted and will be re-executed during the next boot until they also successfully exit)

默认端口及修改方法(可以不改):

- shadowsocks服务端监听端口:1098
 - 文件位置: 服务器/etc/shadowsocks-libev/config.json
 - 如更改, 路由器里 /etc/shadowsocks-libev/config.json也相应更改
- 路由器shadowsocks ss-redir 监听端口:7654
 - 文件位置: 路由器/etc/shadowsocks-libev/config.json
 - 如更改, 路由器/usr/bin/ss-firewall-asia也相应更改
- 路由器shadowsocks ss-tunnel监听端口: 3210
 - 文件位置: 路由器/etc/init.d/shadowsocks
 - 如更改, 路由器 /etc/dnsmasq.d/gfwlist.conf也相应更改

以上端口建议不改。程序运行稳定后, 相关密码可以改掉

端口关联的理解:

- ss-firewall负责把非中国流量转发到本地端口7654
- ss-redir监听端口7654, 该端口流量都加密走自己的服务器通道
- dnsmasq把非国内重要域名的dns查询转发本地3210端口
- ss-tunnel监听本地端口3210,把该端口的dns查询转发到自己服务器向8.8.4.4查询

设置可执行权限







```
chmod +x usr/bin
chmod +x usr/bin/*
chmod +x etc/uci-defaults
chmod +x etc/uci-defaults/defaults
```

相关资源:

- <https://openwrt.org/docs/guide-developer/uci-defaults>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

[下载和设置OpenWrt路由器翻墙配置文件](#)

-  下载翻墙配置文件
-  复制配置文件, 以wndr4300路由器为例:
-  修改配置文件, 编译后就直接可以用了。否则刷上固件后登录路由器再修改。主要修改如下文件:
-  自定义配置文件用途说明
-  关于 /etc/uci-defaults目录
-  设置可执行权限

使用Image Builder编译自动翻墙OpenWrt固件

Image Builder又叫Image Generator, 利用它我们可以方便地定制适合自己无线路由器的固件

编译OpenWrt自定义翻墙固件的注意事项

- 不要用“root”用户编译
- 进入到编译系统目录中执行编译相关命令, 如:~/Downloads/openwrt
- 在编译版的路径中不能够出现空格
- 如果已经用root用户下载并解压了源码, 可用命令改属主成普通用户: sudo chown -R user:user ~/Downloads/openwrt

下载适合自己无线路由器的Image Builder, NetGear WNDR4300 为例

- 进入 <http://downloads.openwrt.org/>
- 选择 Stable Releases或 Development Snapshots
 - 目前的 Stable Releases: <http://downloads.openwrt.org/releases/18.06.1/targets/>
 - Development Snapshots: <http://downloads.openwrt.org/snapshots/targets/>
- 选择 CPU类型, 如 ar71xx: <http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/>
- 选择 Flash 类型, 如nand或generic, 如果是 WNDR4300 路由器, 则选 nand:
<http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/nand/>

下载命令举例:

```
cd ~/Downloads
wget http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/nand/openwrt-imagebuilder-18.06.1-ar71xx-nand.Linux-x86_64.tar.xz
tar -xvf openwrt-imagebuilder-18.06.1-ar71xx-nand.Linux-x86_64.tar.xz
# 为操作方便, 重命名为短目录
mv openwrt-imagebuilder-18.06.1-ar71xx-nand.Linux-x86_64 openwrt-imagebuilder-nand
```

下载包含默认翻墙配置文件的openwrt-fanqiang项目

- git下载openwrt-fanqiang项目

```
cd ~/Downloads
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

- 或者下载zip文件

<https://github.com/softwaredownload/openwrt-fanqiang/archive/master.zip>

本地项目文件夹是: ~/Downloads/openwrt-fanqiang

复制openwrt-fanqiang里面的翻墙配置文件到config-wndr4300目录下

建立一个配置文件夹, 以路由器型号结束, 如 ~/Downloads/config-wndr4300

```
cd ~/Downloads
mkdir config-wndr4300

cd openwrt-fanqiang
cp -R openwrt/default/* ~/Downloads/config-wndr4300/
cp -R openwrt/wndr4300/* ~/Downloads/config-wndr4300/
```

上面的操作, 先复制共用的配置文件 openwrt/default/* 到 config-wndr4300目录下

然后复制WNDR4300专用的配置文件(如果存在)到 openwrt/WNDR4300/* 到 config-wndr4300目录下, 如果有同名文件就覆盖

如果你要贡献本项目, 也是先在openwrt-fanqiang/openwrt目录下先建立路由器型号为名称的文件夹, 再把专用的配置文件放到此文夹下。注意文件夹和文件名都是小写的



修改TL-WNDR4300路由器翻墙配置文件

主要修改以下文件：

```
config-wndr4300/etc/shadowsocks-libev/config.json
config-wndr4300/usr/bin/ss-firewall-asia
config-wndr4300/etc/uci-defaults/defaults
```

为了方便以后升级，可以写个bash文件自动修改配置文件

一切操作尽量自动化，你甚至可以自动化一切操作：下载ImageBuilder，下载OpenWrt源码，下载shadowsocks-libev源码，同步openwrt-fanqiang源码，编译ipk，修改翻墙设置，编译翻墙固件，早上一觉醒来，新鲜出炉、美味可口的翻墙固件就已经摆放在桌上了

下面是一个自动修改配置文件的例子，从中可以知道需要修改哪些地方。从2015年12月起，可能用于自动化修改的默认值都应该标准化，方便自动化操作

```
#!/bin/bash

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date: 2015-12-24

REPOSITORY=~/Downloads/openwrt-fanqiang
CONFIG=~/Downloads/config-wndr4300

createdir() {
    rm -rf $CONFIG
    mkdir $CONFIG
}

copy() {
    cp -R $REPOSITORY/openwrt/default/* $CONFIG/
    cp -R $REPOSITORY/openwrt/wndr4300/* $CONFIG/
}

setmod() {
    chmod +x $CONFIG/usr/bin/*
    chmod +x $CONFIG/etc/uci-defaults
    chmod +x $CONFIG/etc/uci-defaults/*
}

modify() {
    # server ip address
    sed -i 's/1.0.9.8/server_ip/' $CONFIG/etc/shadowsocks-libev/config.json

    # server_port
    sed -i 's/1098/server_port/' $CONFIG/etc/shadowsocks-libev/config.json

    # local_port
    sed -i 's/7654/7654/' $CONFIG/etc/shadowsocks-libev/config.json

    # password
    sed -i 's/killgfw/killgfw/' $CONFIG/etc/shadowsocks-libev/config.json

    # method
    sed -i 's/chacha20-ietf-poly1305/chacha20-ietf-poly1305/' $CONFIG/etc/shadowsocks-libev/config.json

    # local_port
    sed -i 's/7654/7654/' $CONFIG/usr/bin/ss-firewall-asia

    # ppoe username
    sed -i 's/wan-username/wan-username/' $CONFIG/etc/uci-defaults/defaults

    # ppoe password
    sed -i 's/wan-password/wan-password/' $CONFIG/etc/uci-defaults/defaults

    # wifi password
    sed -i 's/icanfly9876/icanfly9876/g' $CONFIG/etc/uci-defaults/defaults

    # root password
    sed -i 's/\\nfanqiang/\\nfanqiang/' $CONFIG/etc/uci-defaults/defaults
}

if [ "$1" = "createdir" ]; then
    createdir
elif [ "$1" = "copy" ]; then
    copy
elif [ "$1" = "setmod" ]; then
    setmod
elif [ "$1" = "modify" ]; then
```



```

        modify
    else
        echo "usage: createdir copy setmod modify"
    fi

```

自动修改翻墙配置文件用法：

```

./config-wndr4300.sh createdir
./config-wndr4300.sh copy
./config-wndr4300.sh setmod
./config-wndr4300.sh modify

```

☹️ 确定OpenWrt无线路由器的PROFILE值

```

cd openwrt-imagebuilder
make info

```

找到自己固件的型号，比如我的是 `NETGEAR WNDR4300v1`，它的PROFILE值是WNDR4300V1。如下图：

```

WNDR4300V1:
NETGEAR WNDR4300v1
Packages: kmod-usb-core kmod-usb2 kmod-usb-ledtrig-usbport

```

🔍 找出默认应该包含进OpenWrt固件的包

基础包：

对于WNDR4300无线路由器来说，可以这样获取：

```

echo $(wget -qO - http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/nand/config.seed | sed -ne 's/^CONFIG_PACKAGE_\([a-z0-9-]*\)=y/\1/ip
')

```

由于 OpenWrt开发非常活跃，不同版本的基础包可能是不同的

2018-09的基础包：

```

libiwininfo-lua liblua liblucihttp liblucihttp-lua libubus-lua lua luci luci-app-firewall luci-base luci-lib-ip luci-lib-jsonc luci-lib-nixio luci-mod-
admin-full luci-proto-ipv6 luci-proto-ppp luci-theme-bootstrap rpcd rpcd-mod-rrdns uhttpd

```

默认包：

运行命令：

```

make info

```

在顶部会列出：

Current Target: "ar71xx (Generic devices with NAND flash)" Default Packages:

```

base-files libc libgcc busybox dropbear mtd uci opkg netifd fstools uclient-fetch logd kmod-gpio-button-hotplug swconfig kmod-ath9k
wpad-mini uboot-envtools dnsmasq iptables ip6tables ppp ppp-mod-pppoe firewall odhcpd-ipv6only odhcp6c

```

所有型号路由器共用包：

```

Default:
  Default Profile
  Packages:

```

```

kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-usb-ledtrig-usbport

```

特定路由器型号专属包,列出在**PROFILE**的下面，对于 **WNDR4300V1**：

```

kmod-usb-core kmod-usb2 kmod-usb-ledtrig-usbport

```

自定义包(**shadowsocks-libev** 后面四个包是依赖)：

```
ipset ipset-dns wget bind-dig iptables-mod-tproxy kmod-ipt-tproxy ip-full stubby dnsmasq-full simple-obfs libmbdts libcares libev  
libsodium shadowsocks-libev
```

- libmbdts libcares libev libsodium shadowsocks-libev

shadowsocks-libev 及依赖, 需要自己编译

- simple-obfs 是 shadowsocks-libev 混淆插件, 需要自己编译
- stubby 可用于 DNS over TLS
- iptables-mod-tproxy kmod-ipt-tproxy ip-full 用于防火墙 UDP 转发
- dnsmasq-full 需要配合 shadowsocks 客户端 ss-tunnel 使用

Dnsmasq 提供 DNS 缓存和 DHCP 服务功能。作为域名解析服务器(DNS), dnsmasq可以通过缓存 DNS 请求来提高对访问过的网址的连接速度。作为DHCP 服务器, dnsmasq 可以为局域网电脑提供内网ip地址和路由

默认的dnsmasq为base版本, 该版本不能对特定的域名地址进行标记操作(因为我们需要对一些特定域名如twitter等进行标记), 改为更加强大的dnsmasq-full

- bind-dig 可以调试域名解析

如果你的openWrt版本是 ATTITUDE ADJUSTMENT, 可能加上iptables-mod-nat-extra包, 如果没安装的话iptables的端口转发会不支持

上述包整合在一起并去重复。简单方法是复制到 Sublime Text, 以空格分隔, 再用正则把空格 替换成 \n, 然后 Edit -> Permute Lines -> Unique

注意, 在编译前要把自己编译的 shadowsocks-libev 及其他要用到的 .ipk 文件放到ImageBuilder的目录下packages

OpenWrt Image Builder的三个命令行参数

- PROFILE 指定设备型号, 此处是 WNDR4300V1
- PACKAGES 指定要编译进固件的包
- FILES 指定要编译进固件的自定义文件, 如网络有关配置文件, 自定义包, 我们放在 ~/Downloads/config-wndr4300 目录下了 要排除的 package 写在最后面, 格式是 -package

开始编译OpenWrt自动翻墙固件

```
cd ~/Downloads/openwrt-imagebuilder-nand  
make image PROFILE=WNDR4300V1 PACKAGES="libiwinfo-lua liblua liblucihttp liblucihttp-lua libubus-lua lua luci luci-app-firewall luci-base luci-lib-  
ip luci-lib-jsonc luci-lib-nixio luci-mod-admin-full luci-proto-ipv6 luci-proto-ppp luci-theme-bootstrap rpcd rpcd-mod-rrdns uhttpd base-files libc  
libgcc busybox dropbear mtd uci opkg netifd fstools ucliclient-fetch logd kmod-gpio-button-hotplug swconfig kmod-ath9k wpad-mini uboot-envtools iptab  
les ip6tables ppp ppp-mod-pppoe firewall odhcpd-ipv6only odhcp6c kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-usb-ledtrig-usbport ipset ipset-dns wge  
t bind-dig iptables-mod-tproxy kmod-ipt-tproxy ip-full stubby dnsmasq-full simple-obfs libmbdts libcares libev libsodium shadowsocks-libev -dnsmas  
q" FILES=~/.Downloads/openwrt-wndr4300
```

注意, 我们已经使用了 dnsmasq-full ,就不需要用 dnsmasq , 用 -dnsmasq 排除, 否则可能会有编译错误

编译好的的固件在ImageBuilder的bin/targets/ar71xx/目录下

然后把编译出的固件刷进路由器, 重启路由器后就能免设置智能翻墙

刷翻墙固件后管理员登录OpenWrt

刷好固件并重启路由器后, 电脑连上无线网络, 然后就可使用密码 fanqiang 登录路由器

- ssh登录openwrt管理路由器:

```
ssh root@192.168.1.1
```

- 浏览器打开192.168.1.1登录

以后玩OpenWrt出问题, 可以重新刷上这个翻墙固件就又可以网上畅行无阻了

相关资源:

- <https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/>
- <https://openwrt.org/docs/guide-user/additional-software/imagebuilder>
- <https://openwrt.org/zh-cn/doc/howto/obtain.firmware.generate>

- <https://openwrt.org/docs/guide-developer/build-system/use-buildsystem>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07
使用Image Builder编译自动翻墙OpenWrt固件

-  编译OpenWrt自定义翻墙固件的注意事项
-  下载适合自己无线路由器的Image Builder, NetGear WNDR4300 为例
-  下载包含默认翻墙配置文件的openwrt-fanqiang项目
-  复制openwrt-fanqiang里面的翻墙配置文件到config-wndr4300目录下
-  修改TL-WNDR4300路由器翻墙配置文件
-  确定OpenWrt无线路由器的PROFILE值
-  找出默认应该包含进OpenWrt固件的包
-  OpenWrt Image Builder的三个命令行参数
-  开始编译OpenWrt自动翻墙固件
-  刷翻墙固件后管理员登录OpenWrt

如何使用别人预编译的OpenWrt翻墙固件 for TP-LINK WR2543N (包含shadowsocks-libev)

如果你的无线路由器和我的一样，也是 TP-LINK wr2543N v1，你不想自己编译固件，那么可以下载我预先编译好的固件，刷好固件好，稍微设置下，就可以自动翻墙

在下载和刷OpenWrt固件前，确保熟悉本教程的前面部分，已经配置好shadowsocks-libev服务端，并能自由进入路由器的安全模式。再次强调，刷机有风险，风险自承担

该固件只是在OpenWrt trunk版加上: luci-ssl wget shadowsocks-libev的最新版，还有翻墙要用到的配置，没有添加其他任何内容

翻墙默认配置

- 教程用到的OpenWrt翻墙配置文件](<https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt>)
- 教程中用到的shadowsocks服务端配置文件

下载OpenWrt固件 for TP-LINK wr2543N

到下面的网址下载: <https://software-download.name/2014/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade-bin-with-shadowsocks/>

下载后保存在Ubuntu: ~/Downloads/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin

复制OpenWrt固件到路由器

```
scp ~/Downloads/openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin root@192.168.1.1:/tmp/
```

登录OpenWrt路由器,并查看文件大小是否正确

```
ssh root@192.168.1.1
root@OpenWrt: cd /tmp/
ls
```

升级OpenWrt固件(不保留原来配置)

```
root@OpenWrt:/tmp# sysupgrade -n openwrt-ar71xx-generic-tl-wr2543-v1-squashfs-sysupgrade.bin
```

路由器重启后，电脑连接到无线网络 eastking-wr2543

ssh登录并修改设置:

```
ssh root@192.168.1.1
```

输入密码 fanqiang 登录

有时会提示错误:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@  WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!  @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
cf:c5:12:34:56:0b:4d:1c:56:48:6a:87:04:cf:b8:83.
Please contact your system administrator.
```

```
Add correct host key in /home/openwrt-fanqiang/.ssh/known_hosts to get rid of this message.
Offending RSA key in /home/openwrt-fanqiang/.ssh/known_hosts:3
  remove with: ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1
RSA host key for 192.168.1.1 has changed and you have requested strict checking.
Host key verification failed.
```

解决办法就是复制并运行提示中的清理命令：

```
ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1
```

以下设置必须修改：

- /etc/shadowsocks-libev/config.json
 - server必须改成你的服务器实际IP
- /etc/config/network
 - wan-username 和 wan-password必改
- /usr/bin/ss-firewall-asia
 - 1.0.9.8必须改成你的服务器实际IP

如果你还改了其他默认值，请自行修改相应文件。不建议修改其他默认值，以提高一次成功率

执行以下命令使修改生效

```
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks start
#root@OpenWrt:~# /etc/init.d/network restart
```

测试一下是否可以在网上畅行无阻了









本教程已经在github开源，欢迎提交改进，报告bug: <https://github.com/softwaredownload/openwrt-fanqiang>

相关资源：

- <https://fanqiang.software-download.name/>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-12-07

如何使用别人预编译的OpenWrt翻墙固件 for TP-LINK WR2543N (包含shadowsocks-libev)

-  翻墙默认配置
-  下载OpenWrt固件 for TP-LINK wr2543N
-  复制OpenWrt固件到路由器
-  登录OpenWrt路由器,并查看文件大小是否正确
-  升级OpenWrt固件(不保留原来配置)
-  路由器重启后, 电脑连接到无线网络 eastking-wr2543
-  ssh登录并修改设置：
-  执行以下命令使修改生效

Ubuntu 编译和使用 shadowsocks Simple-obfs (obfs-server) 混淆翻墙插件

用shadowsocks翻墙，为什么还要用混淆插件

普通用户上网，多数是访问的 <http://kige.com> 或 <https://kige.com> 这样的网址，每个人访问了什么页面，有关方面是一清二楚，毫无秘密可言

用了 shadowsocks 加密访问以后，白脸知道我们访问了海外的某个 IP 地址，并不知道我们通地这个 IP 地址在做什么，比如访问了什么页面，页面上有什么，他们是不知道的

有人推测，如果较多的流量访问海外某 IP 不常用端口，可能会被怀疑，你不浏览网页（他们不知道你是在翻墙浏览网页），这是在干啥呢，难道是不是良民？

于是有人就提出一个混淆流量的设想，把 shadowsocks 加密后的流量混淆一下，白脸喜欢在管理后台偷偷观察我们在网上干什么，加密的数据再混淆一下，白脸在后台看到我们只是在普通的上网，有时打开我们经常上的网站看一下，一个美女也没有，又是一个无趣至极的人！！

Simple-obfs 就是 shadowsocks 的一个混淆流量的插件

Ubuntu 给 shadowsocks-libev 安装 simple-obfs 混淆流量插件

安装环境: Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-34-generic x86_64)

```
sudo apt-get install simple-obfs

Preparing to unpack .../simple-obfs_0.0.5-2_amd64.deb ...
Unpacking simple-obfs (0.0.5-2) ...
Setting up simple-obfs (0.0.5-2) ...
```

打印一下命令行选项：

```
obfs-server --help

-s <server_host>      Host name or IP address of your remote server.
-p <server_port>      Port number of your remote server.
-l <local_port>        Port number of your local server.
-r <addr>:<port>        Forward traffic to this remote server address.
--obfs <http|tls>      Enable obfuscating: HTTP or TLS (Experimental).

[-a <user>]           Run as another user.
[-f <pid_file>]        The file path to store pid.
[-t <timeout>]         Socket timeout in seconds.
[-c <config_file>]     The path to config file.
[-n <number>]          Max number of open files.
[-b <local_address>]   Local address to bind.

[-6]                  Resolve hostname to IPv6 address first.

[-d <addr>]            Name servers for internal DNS resolver.
[--fast-open]          Enable TCP fast open.
                        with Linux kernel > 3.7.0.
[--mptcp]              Enable Multipath TCP on MPTCP Kernel.

[-v]                  Verbose mode.
[-h, --help]          Print this message.
```

Ubuntu server Showdosocks-libev 启用 siimple-obfs 混淆插件

```
kige@ubuntu:~$ cd /etc/shadowsocks-libev
kige@ubuntu:/etc/shadowsocks-libev$ ls
config.json  config-obfs.json
```

查看一下 config.obfs.json 的默认设置：

```
cat config.obfs.json

{
```

```

"server": "127.0.0.1",
"server_port": 8388,
"local_port": 1080,
"password": "veotFuFl",
"timeout": 600,
"method": "chacha20-ietf-poly1305",
"mode": "tcp_and_udp",
"fast_open": true,
"plugin": "obfs-server",
"plugin_opts": "obfs=tls;failover=127.0.0.1:8443;fast-open"
}

```

我们要把 simple-obfs 作为 shadowsocks-libev 的插件使用, shadowsocks-libev 的默认配置文件是 config.json, 所以要把 config.obfs.json 的内容合并到 config.json:

```

# nobfs means not obfsed
sudo cp config.json config.nobfs.json
sudo cp config.obfs.json config.json
sudo vi config.json
# 修改成类似如下值

{
  "server": ["::0", "0.0.0.0"],
  "server_port": 1098,
  "password": "killgfw",
  "timeout": 600,
  "method": "chacha20-ietf-poly1305",
  "mode": "tcp_and_udp",
  "fast_open": true,
  "ipv6_first": true,
  "plugin": "obfs-server",
  "plugin_opts": "obfs=http;fast-open=true"
}

```

其中 server_port, password, method 可以自定义一下

`["::0", "0.0.0.0"]` 意思是让 simple-obfs 服务端监听本地, 优先IPv6

操作系统开启 [TCP fast_open](#) 后才能在 config.json | shadowsocks.json 中设置

```
"fast_open": true
```

TFO开启成功以后, shadowsocks服务端和客户端数据交换的速度会更快一点, 也就是翻墙会更加流畅一些

设置 nginx 反向代理到 obfs-server

假设你的服务端已经安装了 nginx, 并有了默认网站 kige.com, `/etc/nginx/sites-available/kige.com` 是你的网站配置文件

先备份一下原来的网站配置文件:

```

cd /etc/nginx/sites-available
sudo cp kige.com kige.com.nobfs

```

nobfs means not obfsed

到域名管理面板给网站 kige.com 增加一个子域名, 这里是 32.kige.com

```
ping 32.kige.com
```

如果正常, ping 子域名可以看到服务端 IP 和响应时间

在 `/etc/nginx/sites-available/kige.com` 文件中加入一个新的 server 段代表新建的子域名。为了管理方便, 我们不把设置加到既有 server 段中。反向代理主要设置在 `/location {}` 里

首先要理解 反向代理 的概念。我们浏览一个网页, 一般是通过 nginx 把内容传送到我们的计算机上, 这是从服务端 nginx 到我们计算机的数据流动, 这种情况可以视为 正向代理, nginx 充当了信息传递的代理人

反向代理时, nginx 不是向外部的我们传递数据, 而是向内部的一个程序传递数据, 方向是不是反过来了? 在这里, nginx 是向 simple-obfs 的服务端 obfs-server 传递数据

下面开始编辑 `/etc/nginx/sites-available/kige.com`, 添加一个 server 段

```

sudo vi kige.com
# reverse proxy settings in / location field

server {
    listen 80;
    server_name 32.kige.com;
    charset utf-8;
    gzip on;
    keepalive_timeout 120s;
    location / {
        if ($http_upgrade = "") {
            return 301 https://www.kige.com$request_uri;
        }
        proxy_pass http://[::0]:1098;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}

```

注意, nginx 配置是空格比较敏感的, `if (` 中间有个空格

reverse proxy 反向代理用到 nginx 的一个模块, 一般 nginx 版本已经自带, 用法见下面的链接

相关资源:

- <https://www.robberphex.com/2018/08/806>
- <https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/>
- <https://www.digitalocean.com/community/tutorials/understanding-nginx-http-proxying-load-balancing-buffering-and-caching>
- <http://nginx.org/en/docs/http/websocket.html>
- <https://github.com/shadowsocks/simple-obfs/>
- <https://github.com/aa65535/openwrt-simple-obfs>
- <https://usodesu.ga/2018-04-26/OpenWrt-Transparent-Proxy-with-ss-redir/>
- <https://zenandidi.com/archives/1789>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-10-23

Ubuntu 编译和使用 shadowsocks Simple-obfs (obfs-server) 混淆翻墙插件

-  用shadowsocks翻墙, 为什么还要用混淆插件
-  Ubuntu 给 shadowsocks-libev 安装 simple-obfs 混淆流量插件
-  Ubuntu server Showdosocks-libev 启用 siimple-obfs 混淆插件
-  设置 nginx 反向代理到 obfs-server

深刻理解 shadowsocks simple-obfs 流量混淆插件工作原理

nginx 成为翻墙服务端的前台

要正确配置好 simple-obfs 的前提是深刻理解其工作原理

要深刻理解流量混淆插件的工作原理, 前提是对 nginx 在其中起到的作用有正确的认识

在没有启用 imple-obfs前, shadowsocks 服务端 ss-server 站在前台和客户端:ss-local ss-redir ss-tunnel 直接交换数据, 于是 ss-server 就有可能暴露, 被白脸认出来

启用 simple-obfs 流量混淆插件后, 翻墙服务端应该分成二部分:

- 翻墙服务端前台 nginx
- 翻墙服务端后台 obfs-server 和 ss-server

看到了吗, nginx 成了翻墙服务端的重要组成部分, 明白了这点, 你就可能明白了大半

翻墙服务端暴露在外的是 nginx, 众所周知, nginx 是提供 http https服务的, 走的是 TCP 协议, 外部只可能看到 nginx, 不可能看到后面的 ss-server 和 obfs-server, 正是因为这样, 从理论上来说, 提高了翻墙的安全性

翻墙数据交流程

于是我们很容易就得到翻墙数据交流的流程

- nginx 在前台和翻墙客户端交换数据
- 在服务端内部, nginx 和 simple-obfs 服务端 obfs-server 交换数据
- 数据混淆服务端 obfs-server 和加密服务端 ss-server 交换数据

谁在监听什么端口

我们设置了 `"server_port": 1098` 这个 1098 端口是谁在监听的呢

nginx 对外提供 http 服务, 默认监听的是 TCP/80 端口

nginx 接收到外部数据, 如果是反向代理的数据, 就把数据传递给 `"server_port": 1098`, nginx 并不关心谁在 TCP/1098 接收数据

这个 1098 端口是 obfs-server 在监听的, 也就是交由 obfs-server 来处理数据

obfs-server 一个人完成不了处理数据的任务, 还要和 ss-server 合作, ss-server 会在一个随机的 TCP 端口和 obfs-server 交换数据

obfs=http 是什么意思

不能理解成只有访问类似 <http://kige.com> 这样的网站才混淆, 实际上不管你访问的是 http 还是 https, 流量都加密并混淆, 只不过白脸看到的可能是http流量

simple-obfs 只混淆 TCP 数据吗

问:听说 shadowsocks 的 simple-obfs 流量混淆插件只是混淆 TCP 数据, 不混淆UDP数据

为什么?

答:不用问得那么清楚吧, 有的时候朦胧一点不是更好吗:)

obfs-server 处理的数据来自 nginx TCP/80 端口接收到的数据, 决定权在大哥 nginx 那里, obfs-server 就是想要接收 UDP 数据, 也要大哥点头才行呢

如果客户端需要 UDP 协议进行 DNS 查询, 又该如何是好

如果客户端需要 UDP 查询 DNS, 可以使用 dns-forwarder 将其转换为TCP查询.如果要从UDP传递数据, 可以使用不同的本地代理例如kcptun, 或者直接利用 simple-obfs 承载openvpn 数据

🦊 客户端能不能将 DNS 查询请求通过 UDP 发送到服务端, 由服务端进行查询

shadowsocks-libev 服务端启用 simple-obfs 插件后, 默认服务端工作在 TCP 协议

如果需要 ss-server 接收 UDP 数据, 可以在 config.json 中加入

```
"mode": "tcp_and_udp"
```

这个选项目前只能用于配置文件 config.json 方式启动 ss-server

当你指定了 "mode": "tcp_and_udp" 后, ss-server 也会监听、处理 UDP 数据

相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

深刻理解 shadowsocks simple-obfs 流量混淆插件工作原理

- 🐱 nginx 成为翻墙服务端的前台
- 🦊 翻墙数据交流程
- 🏠 谁在监听什么端口
- 🌐 obfs=http 是什么意思
- 😊 simple-obfs 只混淆 TCP 数据吗
- 📺 如果客户端需要 UDP 协议进行 DNS 查询, 又该如何是好
- 🦊 客户端能不能将 DNS 查询请求通过 UDP 发送到服务端, 由服务端进行查询

OpenWrt 路由器编译使用 Simple-obfs for shadowsocks-libev 混淆插件翻墙

这里可以下载编译好的 Simple-obfs for OpenWrt shadowsocks-libev:

<https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>

编译、使用环境:

- 操作系统: Ubuntu 64 bit
- OpenWrt版本: 18.06.1
- 路由器: NetGear WNDR4300

怎样下载OpenWrt DK

您可以下载已编译的SDK, 也可以使用“make menuconfig”命令自行编译

如果编译 ipk 所使用的 OpenWrt 版本和路由器的 OpenWrt 版本不同, 那么可能会有兼容性问题。为了确保兼容, 我们在同一个页面下载 imagebuilder 和 SDK

今天是2018年9月24日, 适合于WNDR4300路由器的 SDK 下载地址为: <http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/nand/>

先决条件:

请参阅[OpenWrt Buildroot](#)页面以安装所需的软件以在SDK上构建软件包

注意:在某些主机上, 需要安装ccache包

Ubuntu 下编译 simple-obfs ipk 详细过程

- 安装 ccache

```
sudo apt-get install ccache
```

- 下载 OpenWrt-SDK

```
cd ~/Downloads
wget http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/nand/openwrt-sdk-18.06.1-ar71xx-nand-gcc-7.3.0_musl.Linux-x86_64.tar.xz

tar -xvf openwrt-sdk-18.06.1-ar71xx-nand-gcc-7.3.0_musl.Linux-x86_64.tar.xz
mv openwrt-sdk-18.06.1-ar71xx-nand-gcc-7.3.0_musl.Linux-x86_64 openwrt-sdk
cd openwrt-sdk-nand
```

- 添加 feeds

```
git clone https://github.com/shadowsocks/openwrt-feeds.git package/feeds
```

- 获取 simple-obfs Makefile

```
git clone https://github.com/aa65535/openwrt-simple-obfs.git package/simple-obfs
```

- [最大化控制台](#), 否则可能有错误

```
include/toplevel.mk:136: recipe for target 'menuconfig' failed
```

- 选择要编译的包 Network -> shadowsocks-libev

```
make menuconfig
```

选择 Network ----> 回车进入, 选择 simple-obfs

- 开始编译

```
make package/simple-obfs/compile V=99
```

把 openwrt-sdk-nand/packages/mips_24kc/base 下的 simple-obfs_0.0.5-3_mips_24kc.ipk 复制到 imagebuilder/packages 目录下就可以编译进翻墙固件

OpenWrt 路由器安装 simple-obfs

```
scp simple-obfs.ipk root@192.168.1:/tmp/
ssh root@192.168.1.1
cd /tmp/
opkg install simple-obfs
```

查看一下 simple-obfs 客户端 obfs-local 参数：

```
obfs-local -h

simple-obfs 0.0.5
maintained by Max Lv

usage:

obfs-local

-s <server_host>      Host name or IP address of your remote server.
-p <server_port>      Port number of your remote server.
-l <local_port>        Port number of your local server.
--obfs <http|tls>      Enable obfuscating: HTTP or TLS (Experimental)

--obfs-host <host_name> Hostname for obfuscating (Experimental).
--obfs-uri <uri_path>   HTTP path uri for obfuscating (Experimental).

[-a <user>]           Run as another user.
[-f <pid_file>]        The file path to store pid.
[-t <timeout>]         Socket timeout in seconds.
[-c <config_file>]     The path to config file.
[-n <number>]          Max number of open files.
[-b <local_address>]   Local address to bind.

[--fast-open]          Enable TCP fast open.
                        with Linux kernel > 3.7.0.
[--mptcp]              Enable Multipath TCP on MPTCP Kernel.

[-v]                  Verbose mode.
[-h, --help]          Print this message.
```

OpenWrt 路由器配置 simple-obfs obfs-local ss-redir

```
# 登录 OpenWrt 路由器
root@192.168.1.1

cd /etc/
cp shadowsocks.json shadowsocks.nobfs.json
vi shadowsocks.json
# 改成类似如下的值：

{
  "server": "32.kige.com",
  "server_port": 80,
  "password": "killgfw",
  "local_port": 7654,
  "method": "chacha20-ietf-poly1305",
  "timeout": 600,
  "fast_open": true,
  "plugin": "obfs-local",
  "plugin_opts": "obfs=http;obfs-host=32.kige.com;fast-open"
}
```

深刻 simple-obfs 客户端 obfs-local 参数用法

- "server": "32.kige.com"

服务端地址, 这里写了域名, 也可以写 IP 地址

- "server_port": 80

还记得吗, 没有使用混淆数据插件 simple-obfs 时, 我们默认的 server_port 是 1098, 那时的1098端口是可以自定义的, 现在的 80 端口是固定的

因为 obfs-local 要把数据发送到服务端 nginx http 服务监听的 80 端口, 如果你写成 87 端口, nginx 是接收不到数据的, 那么客户端和服务就无法交换数据, 何谈翻墙

明显, 这里 obfs-local 在和服务端交换数据时在站前面, shadowsocks-libev 客户端 ss-redir 隐在后面

- "local_port": 7654

这是 ss-redir 在本地监听的端口, 可以自定义, 只要和 /usr/bin/ss-firewall-asia 转发数据的端口一致就可以了

iptables 把需要加密和混淆的数据发到路由器 7654 端口, ss-redir 收到后和再和 obfs-local 一起加密和混淆数据, 再由 obfs-local 发送到 32.kige.com:80

- "method": "chacha20-ietf-poly1305"

升级版数据加密算法, 被检测出数据特征的概率较小

如果用 xchacha20-ietf-poly1305 则更为安全, 是目前最安全的加密算法, 不过可能比 chacha20-ietf-poly1305 消耗更多的计算资源

- "fast_open": true

操作系统开启 TCP fast_open 后才能在 config.json | shadowsocks.json 中设置

"fast_open": true

TFO开启成功以后, 数据交换的速度会更快一点

相关资源:

- <https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07
OpenWrt 路由器编译使用 Simple-obfs for shadowsocks-libev 混淆插件翻墙

-  怎样下载OpenWrt DK
-  Ubuntu 下编译 simple-obfs ipk 详细过程
-  OpenWrt 路由器安装 simple-obfs
-  OpenWrt路由器配置 simple-obfs obfs-local ss-redir
-  深刻simple-obfs 客户端 obfs-local 参数用法

Windows PC翻墙最好方法: shadowsocks-libev + simple-obfs + TFO

shadowsocks Windows客户端有哪些

Shadowsocks 有好几种Windows客户端:

- [Shadowsocks Windows](#)
- [Shadowsocks QT5](#)
- Outline Google 开发, 因为 Google 擅长收集个人隐私, 不推荐使用

shadowsocks-libev 作为资源占用最少的 shadowsocks 实现, 官方却不提供 Windows 预编译可执行文件下载, 虽然他们知道 Windows 用户最多, 但是开发者可能倾向于 Mac Linux 多一点, 也就忽视了 Windows 用户的需求: Windows 用户如果有需要, 可以自己修改编译, 是不是

shadowsocks 官方建议大家在Windows下用 docker, 一个 docker 社区版500多MB, 如果是仅仅为了使用才丁点大的 shadowsocks-libev, 有必要安装这么大的东东吗

还好 [cokebar](#) 自己编译了 Windows 下的 shadowsocks-libev 和 simple-obfs, 下面我们来看看怎么在 Windows PC 上用这两者翻墙

Windows 10 开启 TCP Fast Open

下面的示范包含了 TFO, TFO要操作系统支持才行, 因为翻墙软件要调用操作系统的API

请注意:

TCP Fast Open(TFO)仅适用于Windows 10, 1607或更高版本(精确地, build>= 14393)

怎么知道自己的 Windows 10的版本号?

按 Windows + R键, 输入 `winver` 回车



上图显示版本1803, build 17134.285

如果您使用的是1709(内部版本16299)或更高版本, 则还需要手动开启

- 按 Windows + X
- 选择 Command Prompt(Admin) - 打开控制台(管理员)
- 运行命令:

```
netsh int tcp set global fastopenfallback=disabled
```

- 重启系统以后, 验证 TFO 是否已经开启:

```
netsh interface tcp show global
```

其中 fast open: enabled 说明 TCP fast open 已经启用

```
TCP Global Parameters
-----
Receive-Side Scaling State      : enabled
Receive Window Auto-Tuning Level : normal
Add-On Congestion Control Provider : default
ECN Capability                  : disabled
RFC 1323 Timestamps            : disabled
Initial RTO                     : 3000
Receive Segment Coalescing State : enabled
Non Sack Rtt Resiliency         : disabled
Max SYN Retransmissions         : 2
Fast Open                       : enabled
Fast Open Fallback               : disabled
Pacing Profile                   : off
```

📦 下载 shadowsocks-libev simple-obfs Windows binary可执行文件

<https://software-download.name/2018/shadowsocks-libev-windows-binary-download/>

我下载的是64位版本, 并重命名为 ss-local.exe obfs-local.exe

🏠 shadowsocks-libev + simple-obfs + TFO 服务端设置

🏠 路由器 停止 shadowsocks-libev 翻墙服务

如果你已经按照本教程设置了路由器翻墙, 那么在测试 Windows PC 客户端翻墙前, 首先要停止路由器里面的 shadowsocks-libev

```
root@192.168.1.1
kige@Openwrt:~# /etc/init.d/shadowsocks stop
```

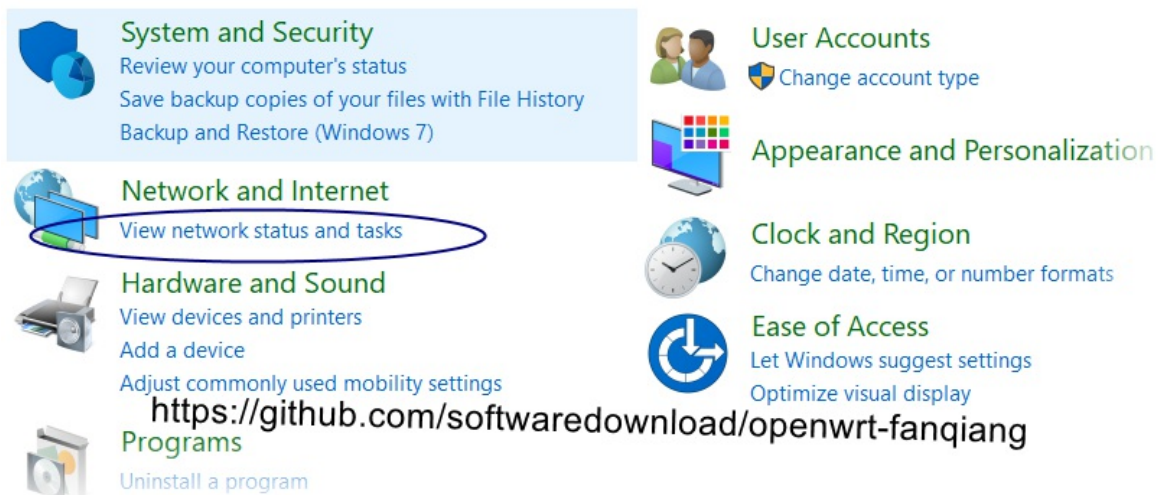
😊 更改网络连接设置

假设你的电脑是 WIFI 上网, 原来通过路由器翻墙时, 我们指定了WIFI连接的静态 IP, 并把DNS设为路由器的内网 IP, 现在要改成动态分配IP, 自动获取DNS

- 按 Windows 键, 输入 control panel 回车打开控制面板
- 选择 View network status and tasks

Adjust your computer's settings

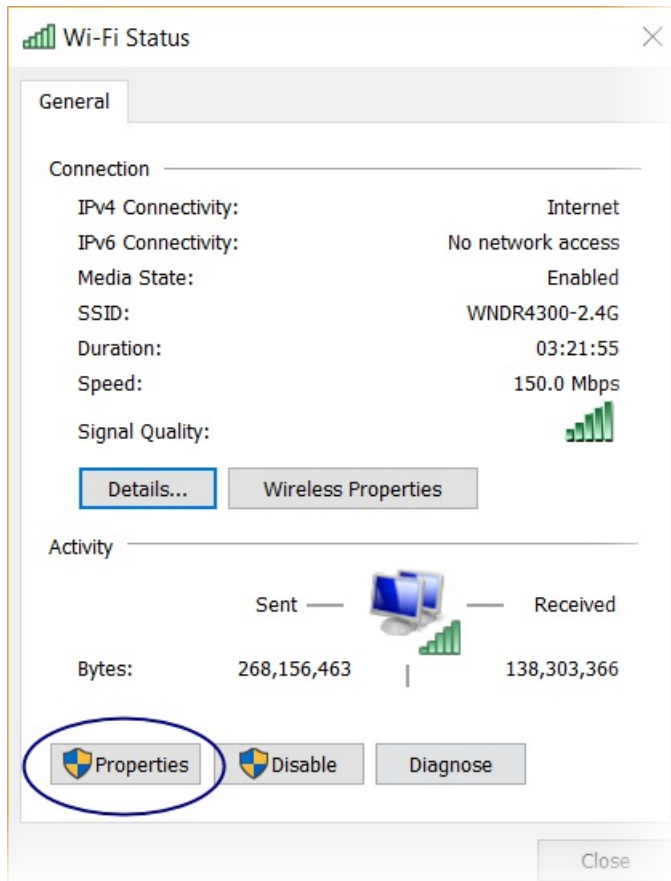
View by:



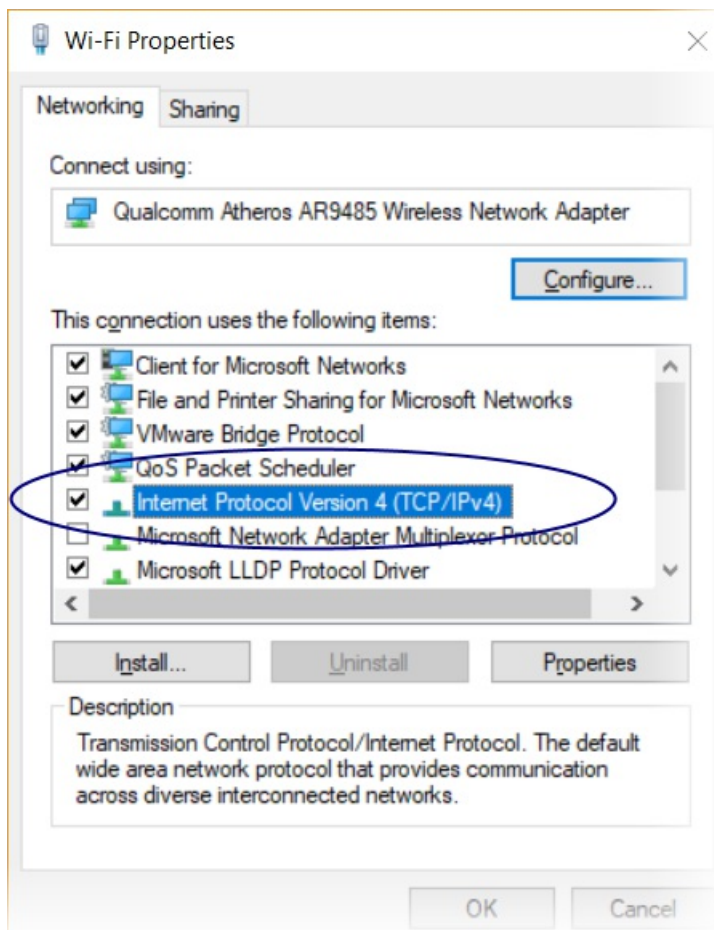
- 点击连上的 WIFI 连接

Access type: Internet
Connections:  Wi-Fi (WNR4300-2.4G)

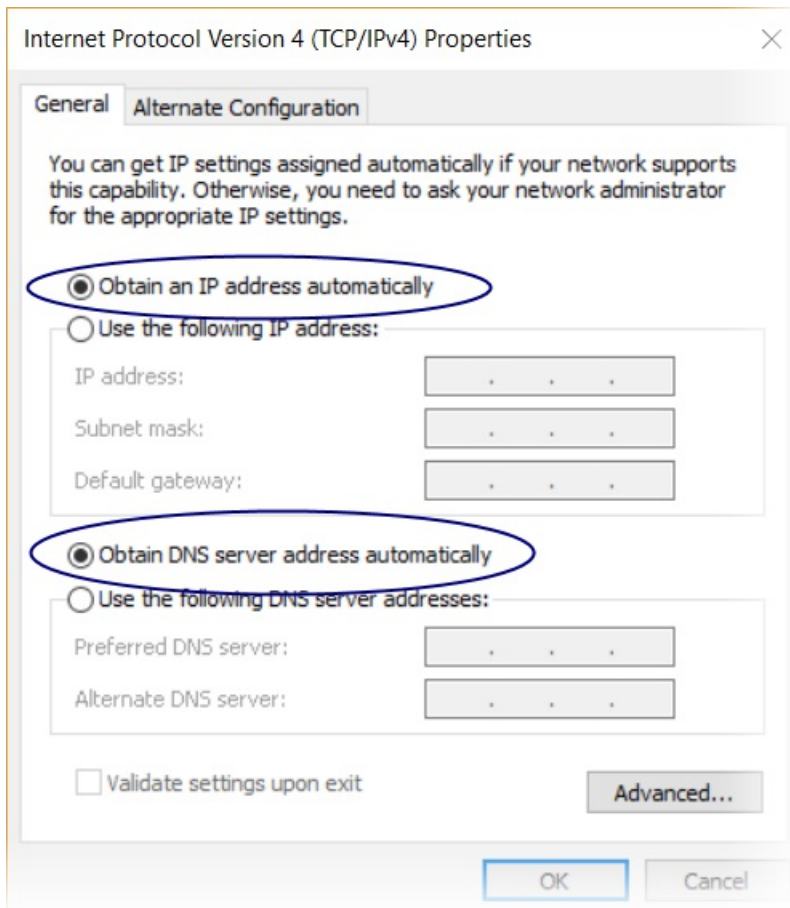
- 点击 Properties (属性)



- 点击 Internet Protocol Version 4(TCP/IPv4)



- 选择 Obtain an IP address automatically(自动获取IP地址), Obtain DNS server address automatically(自动获取DNS服务器地址), OK(确认)



shadowsocks-libev + simple-obfs for Windows 客户端设置

更改网络连接设置后，确认能上国内网站，上不了 <https://youtube.com>

首先创建一个配置文件 shadowsocks.json

```
{
  "server": "1.0.9.8",
  "server_port": 80,
  "password": "killgfw",
  "local_port": 7654,
  "method": "chacha20-ietf-poly1305",
  "timeout": 600,
  "fast_open": true,
  "plugin": "obfs-local",
  "plugin_opts": "obfs=http;obfs-host=32.kige.com;fast-open"
}
```

这个配置文件其实和 [OpenWrt 路由器编译使用 Simple-obfs for shadowsocks-libev 混淆插件翻墙](#) 是一样的，默认把这个文件放在 ss-local.exe obfs-local 同一目录，这样写命令行时可以简短一些

其中 "local_port": 7654 指的是 ss-local 监听本机的 7654 端口，其他软件转发到这个端口的请求都由 ss-local 接收

Windows PC上用 shadowsocks-libev + simple-obfs + TFO 翻墙调试

按 Windows + X，然后选择 Command Prompt 可以打开控制台，默认目录是 C:\Users\your_name> 可以用cd命令进入要操作的目录，或者把文件复制到 C:\Users\your_name> 下

Womdpws 10 修改相应设置后，按住 Shift 再在目录里右击，可以在当前目录打开控制台

建议安装 Git for Windows，安装时选择 OpenSSH，然后在任何目录右击，可以打开当前目录的控制台，这个控制台的操作习惯和 Linux 类似，另外可以和 Linux 下一样的方式操作 ssh

先打印一下 ss-local 的命令行选项，如果我们用了无效的选项，ss-local 会直接退出

```
C:\shadowsocks-libev\64> ss-local -h
shadowsocks-libev 3.2.0
maintained by Max Lv and Linus Yang

usage:

ss-local

-s <server_host>      Host name or IP address of your remote server.
-p <server_port>      Port number of your remote server.
-l <local_port>        Port number of your local server.
-k <password>          Password of your remote server.
-m <encrypt_method>   Encrypt method: rc4-md5,
                      aes-128-gcm, aes-192-gcm, aes-256-gcm,
                      aes-128-cfb, aes-192-cfb, aes-256-cfb,
                      aes-128-ctr, aes-192-ctr, aes-256-ctr,
                      camellia-128-cfb, camellia-192-cfb,
                      camellia-256-cfb, bf-cfb,
                      chacha20-ietf-poly1305,
                      xchacha20-ietf-poly1305,
                      salsa20, chacha20 and chacha20-ietf.
                      The default cipher is chacha20-ietf-poly1305.

[-a <user>]           Run as another user.
[-f <pid_file>]        The file path to store pid.
[-t <timeout>]         Socket timeout in seconds.
[-c <config_file>]     The path to config file.
[-i <interface>]       Network interface to bind.
[-b <local_address>]   Local address to bind.

[-u]                  Enable UDP relay.
[-U]                  Enable UDP relay and disable TCP relay.

[--reuse-port]         Enable port reuse.
[--fast-open]          Enable TCP fast open.
with Linux kernel > 3.7.0.
[--acl <acl_file>]     Path to ACL (Access Control List).
[--mtu <MTU>]          MTU of your network interface.
[--no-delay]           Enable TCP_NODELAY.
[--key <key_in_base64>] Key of your remote server.
[--plugin <name>]       Enable SIP003 plugin. (Experimental)
[--plugin-opts <options>] Set SIP003 plugin options. (Experimental)

[-v]                  Verbose mode.
[-h, --help]          Print this message.
```

然后在命令行输入：

```
ss-local -c shadowsocks.json -v
```

注意，我们启用了 `-v` 以在控制台打印出活动记录，这在调试时十分有用，调试完成后可以去掉这个选项

控制台显示类似下面的信息：

```
C:\shadowsocks-libev\64> ss-local -c shadowsocks.json -v
2018-10-01 11:28:48 INFO: plugin "obfs-local" enabled
2018-10-01 11:28:48 INFO: using tcp fast open
2018-10-01 11:28:48 INFO: initializing ciphers... chacha20-ietf-poly1305
2018-10-01 11:28:48 INFO: listening at 127.0.0.1:7654
2018-10-01 11:28:48 [simple-obfs] INFO: using tcp fast open
2018-10-01 11:28:48 [simple-obfs] INFO: obfuscating enabled
2018-10-01 11:28:48 [simple-obfs] INFO: obfuscation http method: GET
2018-10-01 11:28:48 [simple-obfs] INFO: obfuscating hostname: 32.kige.com
2018-10-01 11:28:48 [simple-obfs] INFO: listening at 127.0.0.1:57373
```

显示 `ss-local` 启用了 `obfs-local` 插件，应用了 `tcp fast open`，并监听在本机 7654 端口。`obfs-local` 应用了 `tcp fast open`，应用了混淆，并监听在本机随机端口和 `ss-local` 通信

这时我们打开浏览器，能翻墙吗？显然不能，因为浏览器并不知道 `ss-local` 监听在本机 7654 端口，更不会把请求发送在那里，怎么可能翻墙呢

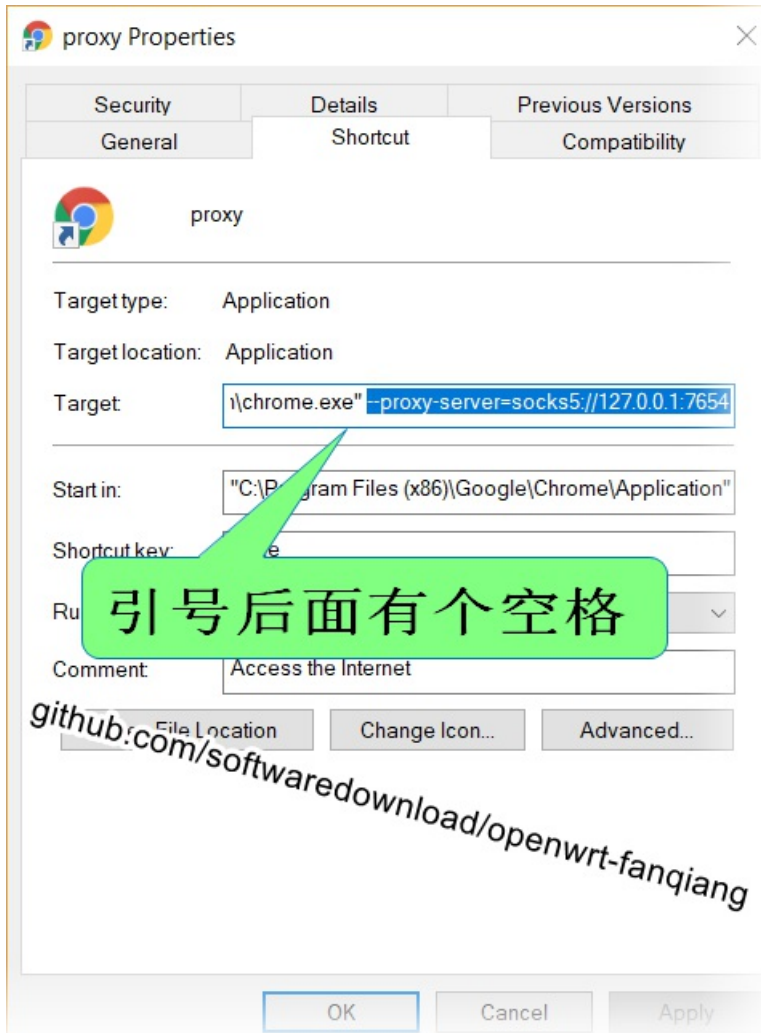
设置 Chrome 浏览器翻墙

接下来我们得告诉浏览器把请求转发到 本机 7654 端口

假设你已经安装了 Chrome 浏览器

- 按 Windows 键, 输入 chrome, 在出来的 Google Chrome 图标上点右键
- 选择 Open File Location 打开文件所在位置
- 这时会打开 Chrome 快捷方式所在文件夹, 并默认选中, Ctrl + C 复制
- 来到桌面, Ctrl + V 粘贴, 把刚粘贴的快捷方式重命名为 Proxy
- 在 Proxy 图标上右击, 选 Properties 属性
- 在 Target (目标)后面加一个英文空格, 再加上下面的内容:

```
--proxy-server=socks5://127.0.0.1:7654
```



设置好以后, 退出已经打开的 Chrome, 点击这个 Proxy(Chrome), 然后 浏览 <https://youtube.com>

如果设置都正确, 应该翻墙无障碍了。这是本浏览器内全局翻墙, 不区分国内、国外 IP, 挺好, 否则打开有些外网会很慢或者根本打不开

这时 DNS 是谁在解析呢? 我们曾在 OpenWrt 路由器里设置 ss-tunnel 把域名解析请求发送到 shadowsocks 服务端, 服务端把解析结果返回到客户端, 从而避免了域名污染

客户端用 ss-local 的时候, 由 ss-local 自动把域名解析请求发送到服务端了, 不需要我们干预。在路由器里也可以用 ss-local 来代替 ss-redir + ss-tunnel + dnsmasq 三者, 不过这样的话国内域名的解析也发到了服务端, 浏览国内网站会有一些延迟

Windows PC 浏览器翻墙最佳方法总结

- 设置 ss-local 随机启动
- 桌面放二个 Chrome 图标, 一个重名为 Proxy 用来翻墙, 另一个命名为 Chrome 不翻墙
- 浏览外网, 关闭打开的 Chrome, 再运行 Proxy
- 浏览内网, 关闭打开的 Proxy, 再运行 Chrome
- 广告屏蔽可以用 host 文件和浏览器插件

shadowsocks-libev + simple-obfs 占用资源极少, 电脑随机启动占用的资源可以忽略不计

本文所述方法是Windows PC浏览器翻墙最佳方法, 适合主要用浏览翻墙的用户。如果是路由器翻墙, 如果打开较多网页, 路由器计算资源有限, 会有压力

如果电脑有较多的需要翻墙的软件, 需要分别设置代理地址, 稍显麻烦, 路由器翻墙会比较方便

相关资源:

- <https://github.com/shadowsocks/simple-obfs/releases/>
- Shadowsocks-libev, simple-obfs for Windows 下载 (支持TFO)
- Shadowsocks-libev, simple-obfs for Windows 下载 (不支持TFO)
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

Windows PC翻墙最好方法: shadowsocks-libev + simple-obfs + TFO

-  shadowsocks Windows客户端有哪些
-  Windows 10 开启 TCP Fast Open
-  下载 shadowsocks-libev simple-obfs Windows binary可执行文件
-  shadowsocks-libev + simple-obfs + TFO 服务端设置
-  路由器 停止 shadowsocks-libev 翻墙服务
-  更改网络连接设置
-  shadowsocks-libev + simple-obfs for Windows 客户端设置
-  Windows PC上用 shadowsocks-libev + simple-obfs + TFO 翻墙调试
-  设置 Chrome 浏览器翻墙
-  Windows PC 浏览器翻墙最佳方法总结

网件Netgear WNDR4300刷OpenWrt翻墙教程

网件Netgear WNDR4300是很多网友推荐的可刷OpenWRT的无线路由器

WNDR4300有v1和v2的区别，目前国行都是v1版本

eastking

StatusSystemNetworkLogout

Status

System

Hostname	eastking
Model	NETGEAR WNDR4300
Firmware Version	OpenWrt Designated Driver r47929 / LuCI (git-15.351.05963-967bb1f)
Kernel Version	4.1.13
Local Time	Tue Dec 22 10:39:19 2015
Uptime	13h 12m 4s
Load Average	0.08, 0.04, 0.05

Memory

Total Available	91060 kB / 125200 kB (72%)
Free	87472 kB / 125200 kB (69%)
Buffered	3588 kB / 125200 kB (2%)

🧐 网件Netgear WNDR4300无线路由器的优点

- 刷OpenWrt方便。购买后，登录管理界面可以直接刷OpenWrt
- WNDR4300自带不死uboot，刷机比较安全
- 硬件配置高。据网友测试，同时接入40台机器都没有问题
- 无线信号强。150平方的室内基本无信号死角
- 有一个 USB 接口，可以挂载设备

🧰 网件Netgear WNDR4300国行硬件信息

千兆双频，300+450Mbps的无线连接，2.4G和5G无线信号可以同时使用，1000Mbps有线端口，内置5天线（两根5G+三根2.4G），采用Atheros AR9344处理器，频率550MHz，128M DDR2内存，128M ROM，USB可接硬盘进行共享，带有wifi开关按钮可以单独关闭无线信号

Version	v1
CPU	Atheros AR9344 rev2 560MHz MIPS 74Kc V4.12
Ram	128MiB
Flash	128MiB NAND
Network	1 WAN + 4x LAN (Gbit)
Wireless	AR9580 [an 3x3:3] + AR9344 [bgn 2x2:2]
USB	Yes

如何购买网件Netgear WNDR4300无线路由器

目前自营电商的价格一般是299元, TB价大约280元

参考信息

<https://openwrt.org/toh/netgear/wndr4300>

- [Netgear WNDR4300 OpenWrt官网Wiki](#)
- [Windows下Netgear WNDR4300刷OpenWrt固件PDF教程 by 书浅](#)
- [预编译集成固件for WNDR4300](#)

最简单的路由器刷OpenWrt翻墙方案:





- <https://github.com/softwaredownload/openwrt-fanqiang>

在线阅读OpenWrt路由器翻墙、科学上网教程:

- <https://fanqiang.software-download.name>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-10-22

[网件Netgear WNDR4300刷OpenWrt翻墙教程](#)

-  [网件Netgear WNDR4300无线路由器的优点](#)
-  [网件Netgear WNDR4300国行硬件信息](#)
-  [如何购买网件Netgear WNDR4300无线路由器](#)
-  [参考信息](#)

下载和设置OpenWrt Image Builder for 网件Netgear WNDR4300路由器

编译详细过程见 [使用Image Builder编译自动翻墙OpenWrt固件](#)

网件Netgear WNDR4300路由器完全使用128M内存教程

将ubi和firmware增加96M, 完全使用128M flash,以实现 WNDR4300路由器 overlay分区大于90MB的功能

在linux下用vi命令可以很方便地查找和修改特定字符

- 查找23552k, 替换成121856k
- 查找25600k, 替换成123904k

下面就用vi来修改:

```
cd ~/Downloads/openwrt-imagebuilder/target/linux/ar71xx/image
cp legacy.mk legacy.mk.bak

vi legacy.mk

#change ubi size to 121856k
# search
/23552k
# delete word
dw
# insert
i
121856k

#change firmware size to 123904k
/25600k
dw
i
123904k

#Save and exit
ZZ
```

更加简单的方法是用 sed 替换:

```
sed -i s/'23552k(ubi),25600k@0x6c0000(firmware)'/ '121856k(ubi),123904k@0x6c0000(firmware)'/ ./legacy.mk
```

修改好后是这样的:

```
wndr4300_mtdlayout=mtdparts=ar934x-nfc:256k(u-boot)ro,256k(u-boot-env)ro,256k(caldata),512k(pot),
2048k(language),512k(config),3072k(traffic_meter),2048k(kernel),121856k(ubi),123904k@0x6c0000(fir
mware),256k(caldata_backup),-(reserved)
```

根据 [此文](#), 修改Flash布局后, 再刷原厂固件, 路由器可能变砖, 请慎重:

对比可以看出来Openwrt做到了和原版一致的Flash分区, 因此正常情况下通过TFTP方式刷官方固件因为分区一致是不会有问题的。
但是如果之前刷入过增加可用空间的改版Openwrt固件的话, 原始的Flash分区就会被破坏

预编译固件都是修改了此二值的

相关资源:

- <https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/ebook/04.3.md>
- <https://kiritox.me/archives/flash-wndr3700v4-to-stock-firmware.html>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05

[下载和设置OpenWrt Image Builder for 网件Netgear WNDR4300路由器](#)

-  网件Netgear WNDR4300路由器完全使用128M内存教程

编译shadowsocks-libev ipk for网件Netgear WNDR4300路由器

不同OpenWrt版本下编译的shadowsocks-libev ipk一般是不能通用的。比如现在用的是trunk版的OpenWrt, 如果使用OpenWrt Chaos Calmer 15.05 下编译的shadowsocks-libev, 可能安装后根本不能启动

前面我曾编译出翻墙固件, 其中shadowsocks-libev是别人编译, 从sourceforge上下载的, 刷上固件后, shadowsocks总是没有自动启动, 运行/usr/bin/ss-redir, 报告没有找到这个文件, 其实文件是在的, 只是不兼容。所以, 最好还是自行编译shadowsocks-libev

按官网的[说法](#), 以下 不要使用root用户来操作

使用SDK编译ipk的新方法教程请参考:[编译shadowsocks-libev for OpenWrt ipk安装包](#) (2018年9月更新)

如果你想节省时间, 建议下载预编译的shadowsocks-libev for OpenWrt ipk安装包:

<https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>

编译shadowsocks-libev ipk安装包(最后更新于2016年)

下面都是在Linux下操作

```
cd ~/Downloads
git clone git://git.openwrt.org/openwrt.git

pushd package
git clone https://github.com/shadowsocks/shadowsocks-libev.git
popd

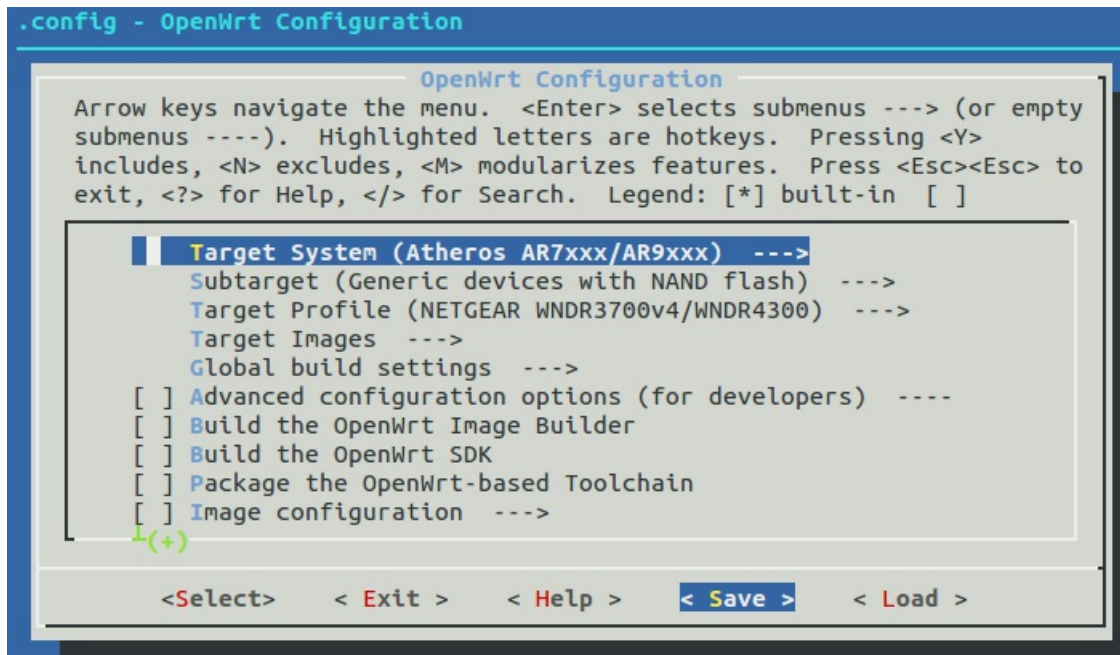
cd ~/Downloads/openwrt
./scripts/feeds update -a
./scripts/feeds install -a

make defconfig
make prereq
make menuconfig

# Target System: Atheros AR7xxx/AR9XXX
# Subtarget: Generic device with NAND flash
# Target Profile: (因我们只是编译包, 这步可以不选)
# Network, 选择shadowsocks-libev-openssl 和 shadowsocks-libev-polarssl, 按m设置为编译独立ipk安装包
# Save && Exit

# 这一步花了几个小时
make tools/install && make toolchain/install

# 开始编译
make V=99 package/shadowsocks-libev/openwrt/compile
```



输出文件在 openwrt/bin/ar71xx/packages/base/目录下, 主要有:

```
shadowsocks-libev_2.4.3_ar71xx.ipk
shadowsocks-libev-polarssl_2.4.3_ar71xx.ipk
libopenssl_1.0.2e-1_ar71xx.ipk
libpolarssl_1.3.15-1_ar71xx.ipk
```

把所有ipk都复制到ImageBuilder的packages目录下

```
cd ~/Downloads/openwrt/bin/ar71xx/packages/base/
cp * ~/Downloads/openwrt-imagebuilder/packages
```

相关资源:

- <https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>
- <https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05

编译shadowsocks-libev ipk for网件Netgear WNDR4300路由器

-  编译shadowsocks-libev ipk安装包(最后更新于2016年)

设置网件Netgear WNDR4300翻墙配置文件

要翻墙成功, 这一步是最重要的

分三步, 下载本项目openwrt-fanqiang; 复制配置文件; 修改配置文件

下面以linux系统 ~/Downloads 下操作为例

下载包含默认翻墙配置文件的openwrt-fanqiang项目

- git下载openwrt-fanqiang项目
`cd ~/Downloads` `git clone https://github.com/softwaredownload/openwrt-fanqiang`
- 或者下载zip文件
<https://github.com/softwaredownload/openwrt-fanqiang/archive/master.zip>

本地项目文件夹是: ~/Downloads/openwrt-fanqiang

复制openwrt-fanqiang里面的翻墙配置文件到openwrt-wndr4300目录下

建立一个配置文件夹, 以路由器型号结束, 如 ~/Downloads/openwrt-wndr4300

```
cd ~/Downloads
mkdir openwrt-wndr4300

cd openwrt-fanqiang
cp -R openwrt/default/* ~/Downloads/openwrt-wndr4300/
cp -R openwrt/wndr4300/* ~/Downloads/openwrt-wndr4300/
```

上面的操作, 先复制共用的配置文件 openwrt/default/* 到 openwrt-wndr4300目录下

然后复制wndr4300专用的配置文件到 openwrt/wndr4300/* 到 openwrt-wndr4300目录下, 如果有同名文件就覆盖

如果你要贡献本项目, 也是先在openwrt-fanqiang/openwrt目录下先建立路由器型号为名称的文件夹, 再把专用的配置文件放到此文夹下。注意文件夹和文件名都是小写的

修改Netgear WNDR4300翻墙配置文件

主要修改以下文件:

```
openwrt-wndr4300/etc/shadowsocks-libev/config.json
openwrt-wndr4300/usr/bin/ss-firewall-asia
openwrt-wndr4300/etc/uci-defaults/defaults
```

为了方便以后升级, 可以写个bash文件自动修改配置文件

一切操作尽量自动化, 你甚至可以自动化一切操作: 下载ImageBuilder, 下载OpenWrt源码, 下载shadowsocks-libev源码, 同步openwrt-fanqiang源码, 编译ipk, 修改翻墙设置, 编译翻墙固件, 早上一觉醒来, 新鲜出炉、美味可口的翻墙固件就已经摆放在桌上了

下面是一个自动修改配置文件的例子, 从中可以知道需要修改哪些地方。从2015年12月起, 可能用于自动化修改的默认值都应该标准化, 方便自动化操作

config-wndr4300.sh:

```
#!/bin/bash

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date: 2015-12-20

REPOSITORY=~/Downloads/openwrt-fanqiang
CONFIG=~/Downloads/openwrt-wndr4300

createdir() {
    rm -rf $CONFIG
    mkdir $CONFIG
}
```

```

}

copy() {
    cp -R $REPOSITORY/openwrt/default/* $CONFIG/
    cp -R $REPOSITORY/openwrt/wndr4300/* $CONFIG/
}

setmod() {
    chmod +x $CONFIG/usr/bin/ss-firewall-asia
    chmod +x $CONFIG/etc/uci-defaults
    chmod +x $CONFIG/etc/uci-defaults/*
}

modify() {
    # server ip address
    sed -i 's/1.0.9.8/server_ip/' $CONFIG/etc/shadowsocks-libev/config.json

    # server_port
    sed -i 's/1098/server_port/' $CONFIG/etc/shadowsocks-libev/config.json

    # local_port
    sed -i 's/7654/7654/' $CONFIG/etc/shadowsocks-libev/config.json

    # password
    sed -i 's/killgfw/killgfw/' $CONFIG/etc/shadowsocks-libev/config.json

    # method
    sed -i 's/chacha20-ietf-poly1305/chacha20-ietf-poly1305/' $CONFIG/etc/shadowsocks-libev/config.json

    # server ip addresss
    sed -i 's/1.0.9.8/server_ip/' $CONFIG/usr/bin/ss-firewall-asia

    # local_port
    sed -i 's/7654/7654/' $CONFIG/usr/bin/ss-firewall-asia

    # ppoe username
    sed -i 's/wan-username/wan-username/' $CONFIG/etc/uci-defaults/defaults

    # ppoe password
    sed -i 's/wan-password/wan-password/' $CONFIG/etc/uci-defaults/defaults

    # wifi password
    sed -i 's/icanfly9876/icanfly9876/g' $CONFIG/etc/uci-defaults/defaults

    # router login password for root
    sed -i 's/\\nfanqiang/\\nfanqiang/' $CONFIG/etc/uci-defaults/defaults
}

if [ "$1" = "createdir" ]; then
    createdir
elif [ "$1" = "copy" ]; then
    copy
elif [ "$1" = "setmod" ]; then
    setmod
elif [ "$1" = "modify" ]; then
    modify
else
    echo "usage: createdir copy setmod modify"
fi

```

config-wndr4300.sh使用方法：

必改值是：

```

server_ip
wan-username
wan-password

```

如果你比较懒, 就改这三项就行了, 可以说本教程是最简单的翻墙方案了

选改值：

```

router login password for root
wifi password

```

其他值一般保持默认值就可以了

假设config-wndr4300.sh在~/Downloads目录下, 运行命令自动修改翻墙配置:




```
cd ~/Downloads
sudo chmod +x config-wndr4300.sh
./config-wndr4300.sh createdir
./config-wndr4300.sh copy
./config-wndr4300.sh setmod
./config-wndr4300.sh modify
```

相关资源:

- <https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05

[设置网件Netgear WNDR4300翻墙配置文件](#)

-  下载包含默认翻墙配置文件的openwrt-fanqiang项目
-  复制openwrt-fanqiang里面的翻墙配置文件到openwrt-wndr4300目录下
-  修改Netgear WNDR4300翻墙配置文件

编译OpenWrt自动翻墙固件 for 网件Netgear WNDR4300路由器

经过前面几个步骤，一切准备就绪，下面就正确开始编译Netgear WNDR4300专用全自动翻墙固件了

编译OpenWrt自动翻墙固件前的系统准备

```
sudo apt-get update
sudo apt-get install git-core build-essential libssl-dev libncurses5-dev unzip
```

OpenWrt Image Builder的三个命令行参数

- PROFILE 指定设备类型, 此处是 WNDR4300V1
- PACKAGES 指定要编译进固件的包
- FILES 指定要编译进固件的自定义文件, 如网络有关配置文件, 默认目录:~/Downloads/openwrt-wndr4300

开始编译OpenWrt自动翻墙固件 for 网件Netgear WNDR4300路由器

命令：

```
cd ~/Downloads/openwrt-imagebuilder
make image PROFILE=WNDR4300V1 PACKAGES="libiwinfo-lua liblua liblucihttp liblucihttp-lua libubus-lua lua luci luci-app-firewall luci-base luci-lib-ip luci-lib-jsonc luci-lib-nixio luci-mod-admin-full luci-proto-ipv6 luci-proto-ppp luci-theme-bootstrap rpcd rpcd-mod-rrdns uhttpd base-files libc libgcc busybox dropbear mtd uci opkg netifd fstools uclient-fetch logd kmod-gpio-button-hotplug swconfig kmod-ath9k wpad-mini uboot-envtools iptables ip6tables ppp ppp-mod-pppoe firewall odhcpd-ipv6only odhcp6c kmod-usb-core kmod-usb2 kmod-usb-ledtrig-usbport luci-ssl ipset ipset-dns wget iptables-mod-nat-extra bind-dig dnsmasq-full libmbdtdls libcares libev libsodium shadowsocks-libev -dnsmasq" FILES=~/Downloads/config-wndr4300
```

编译时报错：

```
opkg_install_cmd: Cannot install package kmod-ipv6
```

移除 kmod-ipv6后编译成功

编译好的的固件在：

```
~/Downloads/openwrt-imagebuilder/bin/targets/
```

其中包含：

```
openwrt-ar71xx-nand-wndr4300-ubi-factory.img
openwrt-ar71xx-nand-wndr4300-squashfs-sysupgrade.tar
```

可见生成了二种格式的固件, img 格式和 tar 格式。其中 img 格式只能用 tftp 的方法进行刷入。而 tar 也只能通过 已刷了Openwrt的WEB端进行刷入

请同时参考[使用Image Builder编译自动翻墙OpenWrt固件](#)

部分编译错误处理：

- Build dependency: Please install the openssl library (with development headers)

For Centos:

```
yum install openssl-devel
```

For Ubuntu:

```
sudo apt-get install libssl-dev
```

- Unable to open feeds configuration in line 42

使用 `svn co svn://svn.openwrt.org/openwrt/trunk/` 下载后再编译的方法没有遇到这个问题

- configure: error: you should not run configure as root (set FORCE_UNSAFE_CONFIGURE=1 in environment to bypass this check)

See config.log' for more details

将下载的文件的所有者改为自己,假设用户名是ubuntu

```
sudo chown -Rv ubuntu /home/ubuntu/openwrt
```




再重新运行 `make`

相关资源:

- <https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05

[编译OpenWrt自动翻墙固件 for 网件Netgear WNDR4300路由器](#)

-  [编译OpenWrt自动翻墙固件前的系统准备](#)
-  [OpenWrt Image Builder的三个命令行参数](#)
-  [开始编译OpenWrt自动翻墙固件 for 网件Netgear WNDR4300路由器](#)

网件Netgear WNDR4300路由器怎样刷OpenWrt自动翻墙固件

🔗 两种翻墙固件格式 img tar的区别

```
openwrt-ar71xx-nand-wndr4300-ubi-factory.img
openwrt-ar71xx-nand-wndr4300-squashfs-sysupgrade.tar
```

我们编译出了两种固件，一种为 ...ubi-factory.img 格式，一种为 ...squashfs-sysupgrade.tar 格式。其中 img 格式只能用 tftp 的方法刷入。而 tar 只能通过已刷了Openwrt的WEB端进行刷入。下面分别说明 两种不同的刷入方法：

tftp刷固件的方式，不管原来的固件是什么格式，都可以刷factory.img

😬 网件Netgear WNDR4300路由器进入恢复模式的方法

- 关闭路由器电源
- 用 牙签, 或其他尖物 按住设备背面的机身背面的红色小圆孔(Restore Factory Settings button)
- 开启电源开关
- 观察电源灯(此时保持按住Restore Factory Settings按钮不要松手)，直到电源灯由 橙色闪烁 状态变到 绿色闪烁 状态(说明设备已经进入到 TFTP修复模式)

🌐 Linux下Netgear WNDR4300路由器用tftp刷翻墙固件

- 将电脑用网线连接到设备的 LAN口，而不是wan口。国行Netgear WNDR4300的wan口是黄色的
- 将电脑的本地连接IP设置为 192.168.1.X (此例中IP地址设置为 192.168.1.9)，子网掩码为 255.255.255.0，网关为192.168.1.1
- 路由器进入恢复模式
- 测试能否连接到路由器：

```
ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
Warning: time of day goes back (-3646479862160196504us), taking countermeasures.
Warning: time of day goes back (-3646479862160196420us), taking countermeasures.
```

- 网件Netgear WNDR4300路由器刷翻墙固件

```
sudo apt-get install tftp
# 进入固件所在目录
cd ~/Downloads/openwrt-imagebuilder/bin/targets/ar71xx/nand
echo -e "binary\nrxtmt 1\ntimeout 60\ntrace\nput openwrt-18.06.1-ar71xx-nand-wndr4300-ubi-factory.img\n" | tftp 192.168.1.1
```

- 观察指示灯，文件会在5秒内传送完毕，等待80秒左右，设备会自动重启(请耐心等待，切勿将路由器手动断电)。设备重启后，看到亮绿灯，一定要按机身后面的电源开关手动断电、开机，否则可能没有无线5G 这不是BUG，其他openwrt也是一样的。每次刷factory.img都要这样
- 路由器完成初始化需要几分钟时间，2.4G 和 5G 的无线信号灯才会亮起，请耐心等待

🐱 Windows下Netgear WNDR4300路由器用tftp刷翻墙固件

- 启用tftp。Windows 10下:控制面板，所有控制面板项，程序和功能，启用或关闭Windows功能，启用“TFTP”客户端
- 将电脑用网线连接到设备的 LAN口
- 将电脑的本地连接IP设置为 192.168.1.X (此例中IP地址设置为 192.168.1.9)，子网掩码为 255.255.255.0，网关192.168.1.1
- 路由器进入恢复模式
- 测试能否连接到路由器：ping 192.168.1.1
- 网件Netgear WNDR4300路由器刷翻墙固件

- 按Windows+R,输入cmd并回车调出命令程序
- 假设openwrt-ar71xx-nand-wndr3700v4-ubi-factory.img在C:\盘

- 运行命令：

```
cd C:\
tftp -i 192.168.1.1 put openwrt-ar71xx-nand-wndr3700v4-ubi-factory.img
```

```
C:\>tftp -i 192.168.1.1 put openwrt-ar71xx-nand-wndr4300-ubi-factory.img
Transfer successful: 6815873 bytes in 3 second(s), 2271957 bytes/s
```

- 观察指示灯, 设备重启后, 看到亮绿灯, 再手动断电、开机, 否则可能没有无线5G

相关资源:

- <https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/>
- <https://openwrt.org/docs/guide-user/installation/generic.flashing.tftp>
- Windows下Netgear WNDR4300刷OpenWrt固件PDF教程 by 书浅

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05
网件Netgear WNDR4300路由器怎样刷OpenWrt自动翻墙固件

-  两种翻墙固件格式 img tar的区别
-  网件Netgear WNDR4300路由器进入恢复模式的方法
-  Linux下Netgear WNDR4300路由器用tftp刷翻墙固件
-  Windows下Netgear WNDR4300路由器用tftp刷翻墙固件

登录并设置已经刷了OpenWrt 翻墙固件的网件Netgear WNDR4300路由器

Netgear WNDR4300 预编译翻墙固件下载(2015-12-22)

<https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/>

你按照**本教程**编译了WNDR4300路由器 OpenWrt 全自动翻墙固件, 并且刷进了路由器, 如果一切正常, 就可以零设置自动翻墙了。运气不够好, 就要登录路由器修改一下设置

你懒得自己编译翻墙固件, 下载了本教程提供的Netgear WNDR4300路由器翻墙固件并刷进了路由器, 就必须手动修改一些值才能自动翻墙

本教程就针对上面这两种情况

怎样登录已经刷了OpenWrt 翻墙固件的网件Netgear WNDR4300路由器

用网线连接电脑和路由器, 将电脑的本地连接IP设置为 192.168.1.97, 子网掩码为 255.255.255.0, 网关为:192.168.1.1

- 网页登录地址: <http://192.168.1.1>
- ssh登录: root@192.168.1.1
- 默认登录密码: fanqiang

Linux下ssh登录WNDR4300路由器并修改设置

```
eastking@ubuntu:~$ ssh root@192.168.1.1
root@192.168.1.1's password:
BusyBox v1.24.1 (2015-12-18 16:02:57 CET) built-in shell (ash)

Author:
https://github.com/softwaredownload/openwrt-fanqiang

# server_ip
root@OpenWrt:~# vi /etc/shadowsocks-libev/config.json

# server_ip
root@OpenWrt:~# vi /usr/bin/ss-firewall-asia

# wan-username, wan-password
root@OpenWrt:~# vi /etc/config/network

# wifi password, optional
root@OpenWrt:~# vi /etc/config/wireless
```

如果你修改了本教程默认的shadowsocks local_port和tunnel_port, 还得修改/etc/dnsmasq.d/下相关文件中的端口号

执行以下命令使修改生效

```
root@OpenWrt:~# /etc/init.d/shadowsocks stop
root@OpenWrt:~# /etc/init.d/shadowsocks start
root@OpenWrt:~# /etc/init.d/dnsmasq restart
root@OpenWrt:~# /etc/init.d/network restart
```

相关资源:

- <https://software-download.name/2015/netgear-wndr4300-openwrt-fanqiang-gujian/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05

[登录并设置已经刷了OpenWrt 翻墙固件的网件Netgear WNDR4300路由器](#)

-  [Netgear WNDR4300 预编译翻墙固件下载\(2015-12-22\)](#)
-  [怎样登录已经刷了OpenWrt 翻墙固件的网件Netgear WNDR4300路由器](#)
-  [Linux下ssh登录WNDR4300路由器并修改设置](#)
-  [执行以下命令使修改生效](#)

D-Link DIR-505路由器刷OpenWrt固件翻墙教程

前面的教程用结合 TP-LINK TL-WR2543N 来讲解翻墙原理与方法, 并不是我特别推荐TP-LINK TL-WR2543N, 而是因为手头正好有这个路由器。毫无疑问, 初学者使用教程同款路由器比较容易上手。但此型号趋向退市, 价格也不便宜, 网上有二手货, 如果功能正常倒也可以考虑

另外的选择, 是使用 D-Link DIR-505 便携式路由器。配置高, 价格便宜

D-Link DIR 505 硬件信息

```
Architecture:   MIPS 24Kc
Vendor:         Atheros
Bootloader:     UBoot 1.1.4
System-On-Chip: SoC: Atheros AR9330 rev 1
CPU/Speed:      Atheros AR9330 400.000MHz
Flash-Chip:     NANYA NT5TU32M16DG-AC
Flash size:     8192 KiB
RAM:            64 MiB
Wireless:       802.11b/g/n
Ethernet:       10/100 full duplex
USB:            Yes 1 x 2.0 ar7240-ehci
Serial:         Yes - tested working over TTL converter (3.3V!)
JTAG:           Nope
```

与之同价格档次的TP-LINK TL-WR706N 150M迷你型无线路由器 AR9331 SOC 2MB Flash/16MB RAM 相比之下简直是垃圾。我花数百元购买的TP-LINK TL-WR2543N, 也不过是8MB Flash, 64MB RAM内存

还有, D-Link DIR-505 自带不死恢复模式, 调试OpenWrt系统出现问题时我们既可以进 D-Link 的恢复模式刷新固件, 也可以进入 OpenWrt 的恢复模式刷新固件, 可谓是最安全的路由器

如何购买 D-Link DIR 505 A1

我不是D-Link的员工, 也无意为其做广告。DIR-505是我购买的第一款D-Link路由器

我是2014年8月从淘宝 D-Link官方旗舰店买的 D-Link DIR 505 A1, 69元, 固件版本号: 1.03CN。买了后, 看了下手机淘宝, 只要59元。准备再入一个, 都刷上 OpenWrt, 方便随时随地无障碍上网

最简单的路由器刷OpenWrt翻墙方案:

- <https://github.com/softwaredownload/openwrt-fanqiang>

在线阅读OpenWrt路由器翻墙、科学上网器教程:

- <https://fanqiang.software-download.name>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-10-22

D-Link DIR-505路由器刷OpenWrt固件翻墙教程

-  D-Link DIR 505 硬件信息
-  如何购买 D-Link DIR 505 A1

如何进入 DIR-505 恢复模式

在学习OpenWrt可能要测试很多配置, 有时会出现错误, 需恢复或补救, 这时就需要进入路由器的恢复模式

有两种方法进入 DIR-505 的恢复模式

进入D-Link 恢复模式

把 DIR-505和计算机用网线连接起来, 设置计算机网卡的IPv4地址为 192.168.0.98, 子网掩码 255.255.255.0, 在路由器启动时顶住 reset 孔, 当红色指示灯开始缓慢闪烁时, 松开reset孔。然后浏览器打开 192.168.0.1, 这里你可以上传原厂固件或刷 OpenWrt 固件

Plug in your computer to the unit, assign it an ip address of 192.168.0.98, and boot the unit up while holding down the reset. Once the red light starts to blink slowly, release the reset, and go to 192.168.0.1 on your web browser. From there you can upload a new image. After successful flashing, you'll see a "Success" page in your browser.

刷新固件完成后, 重新改回自动获取IP地址

进入 OpenWrt 恢复模式

用网线将路由器和电脑连接起来, 将电脑网卡的IPv4地址设置成 192.168.1.97

路由器插上电源重新开机, 在启动时多次按压路由器侧面的圆形 WPS 按钮直到 LED 指示灯开始快速闪烁

For the generic failsafe mode you can follow <https://openwrt.org/zh-cn/doc/howto/generic.failsafe> You can use the WPS button for that. While booting up, just press it several times until the LED flashes very quick. If you're still not able to telnet it on 192.168.1.1 maybe there's something wrong on the client-side.

接下来就是ubuntu 里 telnet 进入 OpenWrt 并设置 root 密码

```
telnet 192.168.1.1
```

telnet连上后就设置root密码, 自动启用 ssh:

```
root@OpenWrt:/# passwd
Changing password for root
New password:
Retype password:
Password for root changed by root
root@OpenWrt:/#
```

可以在 Ubuntu 里 Ctrl + Shift + t 新开一个命令行窗口, 尝试 ssh 连接OpenWrt:

```
ssh root@192.168.1.1
```

如果 ssh 连上了, 则后面设置的内容和 前面 TLWR-2543N 翻墙教程一样了

要注意的是, D-Link DIR-505 使用接口名称 eth1 而不是通常的 eth0

Other than similar routers (e.g., the TP-Link TL-WR703N), the D-Link DIR-505 uses the interface eth1 rather than eth0. This means that if you build your own firmware, you must configure /etc/config/network accordingly (option ifname 'eth1'), or you will not be able to connect later on via Ethernet.

如果 telnet 连不上, 尝试一下直接ssh登录

设置D-Link DIR-505k路由器无线连接

在没有设置无线连接前, 要登录OpenWrt, 必须用网线把电脑和路由器连接起来, 不太方便。设置无线连接后, 电脑就可以通过无线方式连上路由器, 再登录 DIR-505 OpenWrt 进行设置

```
uci set wireless.@wifi-device[0].disabled=0;
uci set wireless.@wifi-iface[0].ssid='eastking-dir505';
uci set wireless.@wifi-iface[0].encryption='psk2+ccmp';
uci set wireless.@wifi-iface[0].key='icanfly9876';
uci commit wireless;
```

设置好无线连接后, 就可以拔掉电脑的有线连接, 连接无线, 再ssh登录路由器

相关资源:

- <https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>
- <https://forum.openwrt.org/viewtopic.php?id=38742&p=8>
- <https://openwrt.org/toh/d-link/dir-505>
- <https://my.oschina.net/umu618/blog/271630>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05
[如何进入 DIR-505 恢复模式](#)

-  [进入D-Link 恢复模式](#)
-  [进入 OpenWrt 恢复模式](#)
-  [设置D-Link DIR-505k路由器无线连接](#)

D-Link DIR-505 A1 刷 OpenWrt固件过程

D-Link 路由器是锁区的, 不能直接刷OpenWrt 固件。要先到D-Link 官方国际站下载原厂固件, 用16进制编辑器把DEF改成CN, 升级固件, 再刷OpenWrt固件

📁 下载D-Link DIR-505 A1 国际版官方固件

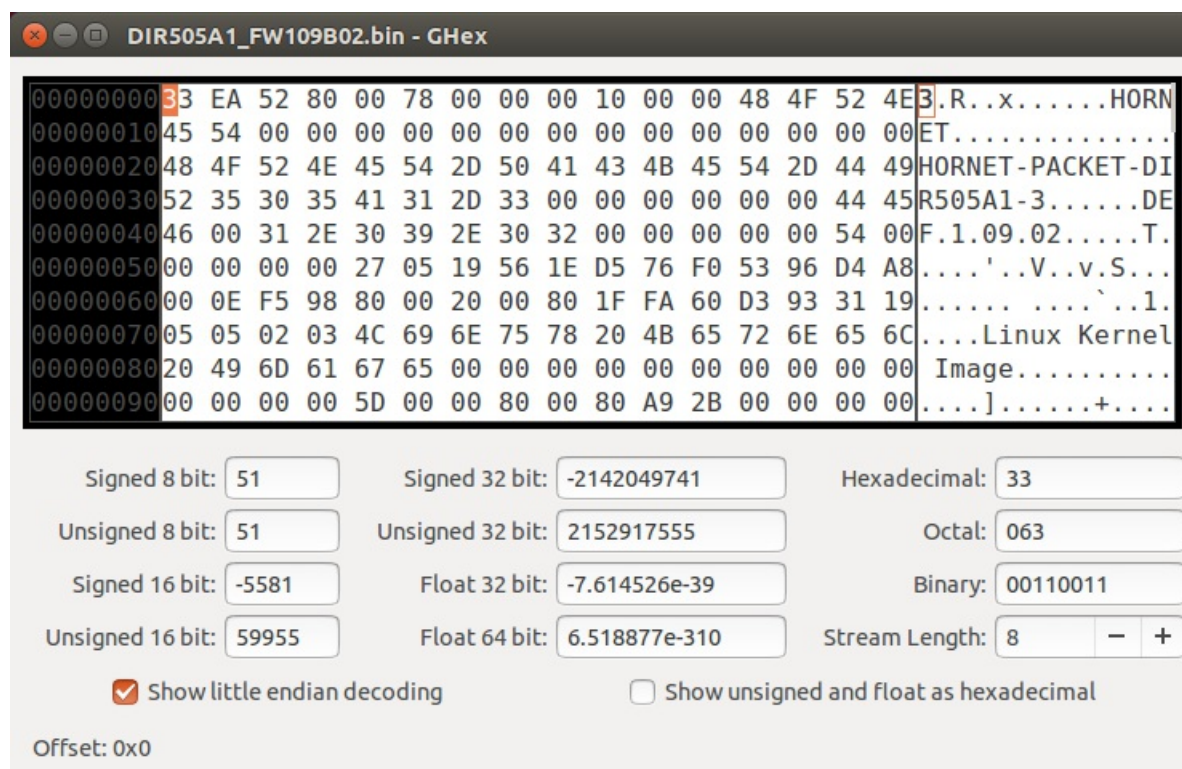
<https://openwrt.org/toh/d-link/dir-505>

下载地址:

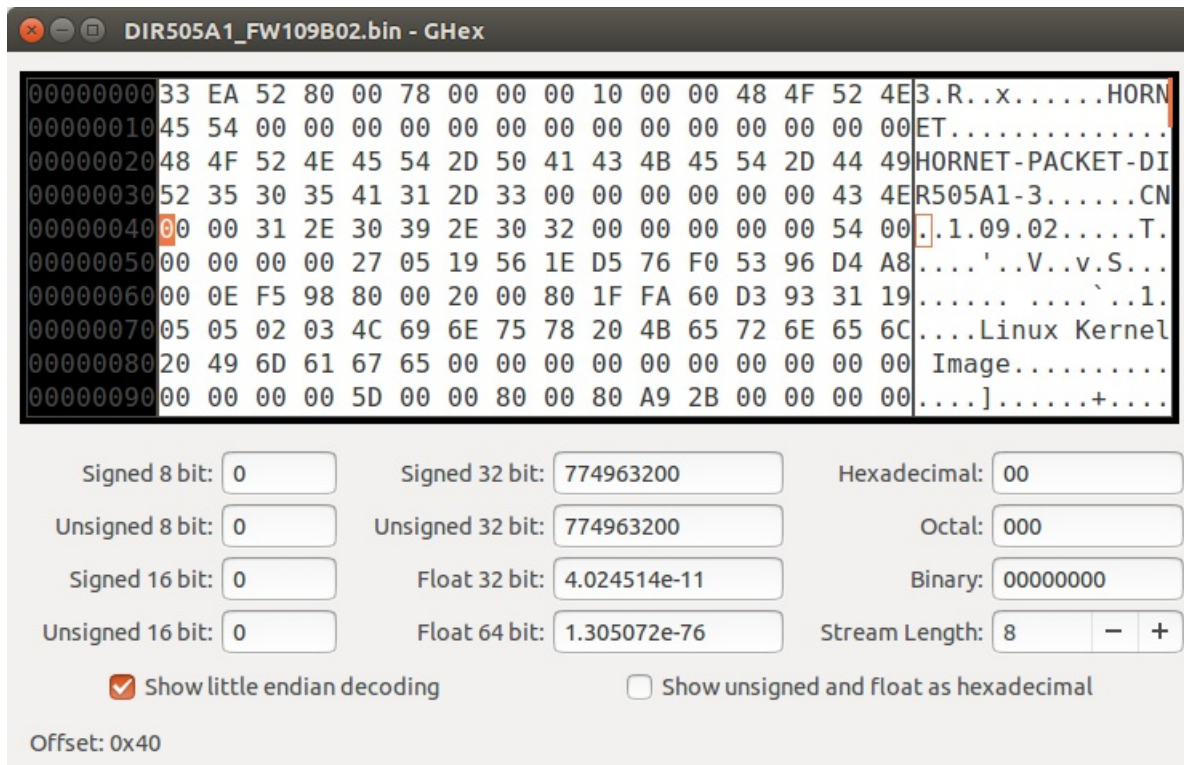
- <http://support.dlink.com.au/download/download.aspx?product=DIR-505>
- ftp://files.dlink.com.au/products/DIR-505/REV_A/Firmware/

😁 用16进制编辑器修改固件的国家代码, DEF 改成 CN

准备一个16进制编辑器, 在本文中, 我用的是Ubuntu下的轻量级16进制编辑器GHex,把固件拖到GHex打开固件



修改后变成如下:



Alt + S 保存对固件的修改

你也可以到下面网址直接下载修改好16进制值的固件：

<https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

👹 刷修改国家后的官方固件

按照路由器官方手册，电脑连上路由器

在 Ubuntu 下电脑连接 DIR-505 路由器的方法：

DIR-505 路由器出厂默认设置没有开启 DHCP，所以我们要给电脑手动设置和路由器同网段的 IPv4 地址才能连上路由器

路由器插上电源。右上角无线信号处，选择 Edit Connections，选择dlink-xxxx，xxxx为路由器MAC ID 的后4位，Edit...，IPv4 Setings, Method选择 Manula 手动，Address 选择 Add，设置：

- Address: 192.168.0.9
- Netmask: 255.255.255.0
- Gateway: 192.168.0.1

如此设置好后电脑就能连上无线网络dlink-xxxx了

浏览器首次进入 <http://192.168.0.1> 会出现设置向导，点取消，然后会出现密码登录页面：



直接点击 登入 按钮, 再点击界面上部的 维护, 然后点击左侧栏的 固件 进入升级固件页面, 点击 **Browse...** 上传我们修改好的固件:



然后点击 上传 按钮完成刷新固件, 接下来就可以刷 OpenWrt固件了

DIR-505A1 刷 OpenWrt 固件

下载 OpenWrt 固件 for DIR-505A1:

- <http://downloads.openwrt.org/snapshots/targets/ar71xx/generic/>
- <http://downloads.openwrt.org/snapshots/targets/ar71xx/generic/openwrt-ar71xx-generic-dir-505-a1-squashfs-factory.bin>

DIR-505刷OpenWrt固件:

我们是在原厂固件上刷 OpenWrt, 一定要下载 factory.bin. 上传后, 等待150秒, DIR-505A1 成功刷上了 OpenWrt 开源固件

固件和语言包信息

当前固件版本 : 1.09

日期 : 2014/June/10

当前语言包版本 : 1.00 CN

日期 : 2012/5/8

网上查询产品固件和语言包最新版本 :

韧体升级

注意: 某些固件升级会将设置复位至出厂默认设置。在进行升级前, 请确认从[维护 - 系统](#)界面保存当前配置

如要升级固件, 您的计算机必须以有线方式接入AP, 输入升级固件的文件名, 然后点击上传按钮。


上传 :

openwrt-ar71xx-generic-dir-505-a1-squashfs-factory.bin

相关资源:

- <https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>
- <https://my.oschina.net/umu618/blog/268466>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05
D-Link DIR-505 A1 刷 OpenWrt 固件过程

-  下载D-Link DIR-505 A1 国际版官方固件
-  用16进制编辑器修改固件的国家代码, DEF 改成 CN
-  刷修改国家后的官方固件
-  DIR-505A1 刷 OpenWrt 固件

D-Link DIR-505启用工作模式开关

DIR-505 硬件开启四种应用模式

D-Link DIR-505 在全球销售多款型号, 不同型号外观不一样, 但内部硬件是一样的。在中国销售的 DIR-505 A1, 也就是本教程所用的型号, 模式开关共有三档, 在开关处动动手, 就可以启用四种模式

撕掉标贴, 去掉螺丝, 就可以打开DIR-505,把开关剪短, 剪掉挡住开关上推的底面, 完工后如下图:



Router模式和AP模式

便携式无线路由器常有Router模式和AP模式, 有的路由器用两个档位对应这两种模式, 拨到Router档就用Router模式, 拨到AP档就用AP模式。DIR-505 原厂固件, Router和AP共用一个档位, 需要用哪种, 需要登录路由器进行选择和设置。现在我们已经刷了 OpenWrt, 档位对应的模式需要自己定义设置

在本教程中, 把新开的第四档作为AP档, 原来的Router/AP档作为Router档

在Router模式时, DIR-505作为无线路由器使用, 有线接口作为WAN口, 连接到ADSL Modem。计算机通过无线的方式连接到路由器。在这种模式下一般需要设置拨号上网帐号

在AP模式时, 通常在DIR-505前端还有路由器, DIR-505的有线接口作为LAN口使用, 前端路由器的LAN口引出网线连接到DIR-505. 在宾馆上网, 把有线扩展为无线常应用此种模式

/etc/rc.local 利用 GPIO 读取开关位置

rc.local内容如下:

```
# Put your custom commands here that should be executed once
# the system init finished. By default this file does nothing.

if [ ! -f /etc/config/backup/network ]; then
    cp /etc/config/network /etc/config/backup/
    cp /etc/config/wireless /etc/config/backup/
    cp /etc/config/firewall /etc/config/backup/
    cp /etc/config/dhcp /etc/config/backup/
fi

read_gpio() {
    (echo $1 > /sys/class/gpio/export) >& /dev/null
    (echo "in" > /sys/class/gpio/gpio$1/direction) >& /dev/null
    return `cat /sys/class/gpio/gpio$1/value`;
}

read_gpio 19;
v=$?;
read_gpio 20;
v=$v$?;
read_gpio 21;
v=$v$?;
read_gpio 22;
v=$v$?;
read_gpio 23;
v=$v$?;
case "$v" in
    10001) v="router";;
    11001) v="repeater";;
    01001) v="hotspot";;
    11000) v="ap";;
    *) v="error";;
esac

/usr/bin/$v

logger working mode: $v

exit 0
```

原理:先备份原始的配置文件,不同模式的设置都是基于原始配置文件,避免出现混乱

在/usr/bin目录下创建相应模式的bash文件,根据不同的GPIO值调用的不同的文件 在本教程中主要应用 /usr/bin/router和 /usr/bin/ap这两个文件

代码的最新版本, 请查看:

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/dir505>

你使用时,可以直接下载整个项目到本地,所有配置文件自然在其中:




```
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

相关资源:

- <https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>
- <https://my.oschina.net/umu618/blog/273945>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05

[D-Link DIR-505启用工作模式开关](#)

-  [DIR-505 硬件开启四种应用模式](#)
-  [Router模式和AP模式](#)
-  [/etc/rc.local 利用 GPIO 读取开关位置](#)

DIR-505 Router 模式



/usr/bin/router 代码:

```
#!/bin/sh

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date: 2014-08-22

cp /etc/config/backup/* /etc/config/

uci delete network.lan.ifname
uci delete network.lan.type

uci add network interface
uci rename network.@interface[-1]='wan'
uci commit network

uci set network.wan.ifname='eth1'
uci set network.wan.peerdns=0
uci set network.wan.proto='pppoe'
uci set network.wan.username='wan-username'
uci set network.wan.password='wan-password'
uci set network.wan.peerdns=0

uci commit network

# default is no this option
#uci set dhcp.lan.ignore=0
#uci commit dhcp

uci set wireless.@wifi-device[0].channel=11
uci set wireless.@wifi-device[0].txpower=15
uci set wireless.@wifi-device[0].disabled=0
uci set wireless.@wifi-device[0].country='CN'
uci set wireless.@wifi-iface[0].mode='ap'
uci set wireless.@wifi-iface[0].ssid='eastking-dir505'
uci set wireless.@wifi-iface[0].encryption='psk2'
uci set wireless.@wifi-iface[0].key='icanfly9876'

uci commit wireless
wifi

/etc/init.d/network restart
```



代码说明:

先把备份的原始配置文件覆盖到配置文件目录, 所有设置都基于原始配置文件 在使用Router 模式时, 有线接口为WAN口, 这时wan的 interface name 是 eth1, 默认lan的interface name 使用了 eth1, 要删除

相关资源:

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/dir505>
- <https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05
DIR-505 Router 模式

- /usr/bin/router 代码:
- 代码说明:

DIR-505 AP 模式翻墙教程



/usr/bin/ap 代码:

```
#!/bin/sh

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date: 2014-08-22

cp /etc/config/backup/* /etc/config/

uci set network.lan.gateway=192.168.1.1
uci set network.lan.dns=192.168.1.1
uci set network.lan.ipaddr=192.168.1.97

uci commit network

uci set dhcp.lan.ignore=1
uci commit dhcp

uci set wireless.@wifi-device[0].channel=11
uci set wireless.@wifi-device[0].txpower=15
uci set wireless.@wifi-device[0].disabled=0
uci set wireless.@wifi-device[0].country='CN'
uci set wireless.@wifi-iface[0].mode='ap'
uci set wireless.@wifi-iface[0].ssid='eastking-dir505'
uci set wireless.@wifi-iface[0].encryption='psk2'
uci set wireless.@wifi-iface[0].key='icanfly9876'

uci commit wireless
wifi

/etc/init.d/network restart
```



代码说明:

在AP模式下, DIR-505的有线接口作为LAN口使用, 连接到前端路由器的LAN口 假设DIR-505前端路由器的IP地址是192.168.1.1, 设置DIR-505的lan 网关和dns都是192.168.1.1, 再设置 DIR-505的 lan IP地址为192.168.1.97



DIR-505穿越功夫网翻墙方法

假设上级路由器没有设置翻墙:

电脑设置无线连接 eastking-dir505: IPv4地址是 192.168.1.53(不同于路由器的地址), 设置子网掩码为255.255.255.0, 网关和DNS为路由器的地址即192.168.1.97, 重启路由器后, 电脑连上 eastking-dir505 即可自动翻墙

原理: 以DIR-505作为DNS服务器, 我们已经把DIR-505设置成翻墙路由器, 自然可以打败功夫网了

假设上级路由器已经翻墙:

电脑设置无线连接 eastking-dir505为DHCP即可 原理: 以上级路由器作为DNS服务器, 上级路由器已经翻墙, 二级路由器就可以免设置自动翻墙了

如果你想节省路由器资源, 这时就可以禁用 dir-505 dns及翻墙相关服务:

```
/etc/init.d/dnsmasq stop
/etc/init.d/dnsmasq disable
/etc/init.d/shadowsocks stop
/etc/init.d/shadowsocks disable
```

代码的最新版本, 请查看:

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/dir505>

相关资源:

- <https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>
- <https://openwrt.org/docs/guide-user/network/wifi/bridgedap>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name>
DIR-505 AP 模式翻墙教程

2018-12-05

-  /usr/bin/ap 代码:
-  代码说明:
-  DIR-505穿越功夫网翻墙方法

编译OpenWrt全自动翻墙固件 for D-Link DIR-505 A1

除了增加模式转换开头, 其他和编译 NetGear WNDR4300翻墙固件一样

你也可以直接下载编译好的翻墙固件: <https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

下载适合D-Link DIR505无线路由器的Image Builder

Image Builder又叫Image Generator, 利用它我们可以方便地定制适合自己无线路由器的固件

选择 **OpenWrt**版本::

- 进入 <http://downloads.openwrt.org/>
- Stable Release, 最后发行的稳定版本:
 - OpenWrt 18.06.1
 - Released: Sat, 18 Aug 2018
- 进入 <http://downloads.openwrt.org/releases/18.06.1/targets/>

选择 **CPU** 类型::

选择 **ar71xx**: <http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/>

选择 **Flash** 类型::

选择 **generic**:

<http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/generic/>

下载 **Image Builder for DID-505**:

- 页面搜索 dir-505 找到适合你的版本号, 比如我的是 dir-505-a1
- 下载 Image Builder:

```
cd ~/Downloads
wget http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/generic/openwrt-imagebuilder-18.06.1-ar71xx-generic.Linux-x86_64.tar.xz
tar -xf openwrt-imagebuilder-18.06.1-ar71xx-generic.Linux-x86_64.tar.xz
mv openwrt-imagebuilder-18.06.1-ar71xx-generic.Linux-x86_64 openwrt-imagebuilder-generic
```

确定OpenWrt无线路由器的PROFILE值

```
cd openwrt-imagebuilder-generic
make info
```

找到自己固件的型号, D-Link DIR 505 A1的PROFILE值是DIR505A1。如下图:

```
DIR505A1:
  D-Link DIR-505 rev. A1
  Packages: kmod-usb-core kmod-usb2 kmod-ledtrig-usbdev
DIR600A1:
  D-Link DIR-600 rev. A1
```

找出默认应该包含进OpenWrt固件的包

对于D-Link DIR-505 A1 无线路由器来说, 可以这样获取:

```
echo $(wget -qO - http://downloads.openwrt.org/releases/18.06.1/targets/ar71xx/generic/config.seed | sed -ne 's/^CONFIG_PACKAGE_\([a-z0-9-]*\)=y/\1\n/ip')
```

2018-09的基础包:

```
libiwinfio-lua liblua liblucihttp liblucihttp-lua libubus-lua lua luci luci-app-firewall luci-base luci-lib-ip luci-lib-jsonc luci-lib-nixio luci-mod-admin-full luci-proto-ipv6 luci-proto-ppp luci-theme-bootstrap rpcd rpcd-mod-rrdns uhttpd
```

默认包:

运行命令：

```
make info
```

在顶部会列出：

Current Target: "ar71xx (Generic devices with GENERIC flash)" Default Packages:

```
base-files libc libgcc busybox dropbear mtd uci opkg netifd fstools uclient-fetch logd kmod-gpio-button-hotplug swconfig kmod-ath9k
wpad-mini uboot-envtools dnsmasq iptables ip6tables ppp ppp-mod-pppoe firewall odhcpd-ipv6only odhcp6c
```

所有型号路由器共用包：

```
Default:
Default Profile
Packages:
```

```
kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-usb-ledtrig-usbport
```

特定路由器型号专属包,列出在**PROFILE**的下面, 对于 **DIR505A1**：

```
kmod-usb-core kmod-usb2 kmod-usb-ledtrig-usbport
```

自定义包：

```
wget bind-dig iptables-mod-tproxy kmod-ipt-tproxy ip-full dnsmasq-full simple-obfs libmbedtls libcares libev libsodium shadowsocks-libev
```

- libmbedtls libcares libev libsodium shadowsocks-libev

shadowsocks-libev 及依赖, 需要自己编译, 或下载编译好的包：

<https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>

- simple-obfs 是 shadowsocks-libev 混淆插件, 需要自己编译, 或下载编译好的包：

<https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>

- iptables-mod-tproxy kmod-ipt-tproxy ip-full 用于防火墙 UDP 转发
- dnsmasq-full 需要配合 shadowsocks 客户端 ss-tunnel 使用

Dnsmasq 提供 DNS 缓存和 DHCP 服务功能。作为域名解析服务器(DNS), dnsmasq可以通过缓存 DNS 请求来提高对访问过的网址的连接速度。作为DHCP 服务器, dnsmasq 可以为局域网电脑提供内网ip地址和路由

默认的dnsmasq为base版本, 该版本不能对特定的域名地址进行标记操作(因为我们需要对一些特定域名如twitter等进行标记), 改为更加强大的dnsmasq-full

- bind-dig 可以调试域名解析

上述包整合在一起并去重复。简单方法是复制到 Sublime Text, 以空格分隔, 再用正则把空格 替换成 \n, 然后 Edit -> Permute Lines -> Unique

注意, 在编译前要把自己编译的 shadowsocks-libev 及其他要用到的 .ipk 文件放到ImageBuilder的目录下packages



按照教程 编译shadowsocks-libev for OpenWrt ipk安装包



下载和设定自定义翻墙配置文件

下面以linux系统 ~/Downloads 下操作为例

```
cd ~/Downloads
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

本地项目文件夹是：~/Downloads/openwrt-fanqiang

建立一个配置文件夹, 以路由器型号结束, 如 ~/Downloads/openwrt-dir505

```
cd ~/Downloads
mkdir openwrt-dir505
cd openwrt-fanqiang
```

```
cp -R openwrt/default/* ~/Downloads/openwrt-dir505/
cp -R openwrt/dir505/* ~/Downloads/openwrt-dir505/
```

上面的操作, 先复制共用的配置文件 openwrt/default/* 到 openwrt-dir505目录下

然后复制dir505专用的配置文件到 openwrt/dir505/* 到 openwrt-dir505目录下, 如果有同名文件就覆盖

设置可执行权限

```
cd ~/Downloads/openwrt-dir505
chmod +x usr/bin
chmod +x usr/bin/*
chmod +x etc/uci-defaults
chmod +x etc/uci-defaults/*
```

说明: etc/uci-defaults目录下的文件会在路由器第一次启动时执行一次。在这里我们设置一些常用值

必须修改的DIR505翻墙配置文件:

- ~/Downloads/openwrt-dir505/etc/shadowsocks-libev/config.json
server改成你的服务器实际IP
- ~/Downloads/openwrt-dir505/usr/bin/router
wan-username 和 wan-password改成实际值
- ~/Downloads/openwrt-dir505/usr/bin/ss-firewall-asia
1.0.9.8必须改成你的服务器实际IP

自动复制和修改DIR-505翻墙设置文件

config-dir505.sh:

```
#!/bin/bash

# Author: https://github.com/softwaredownload/openwrt-fanqiang
# Date: 2015-12-24

REPOSITORY=~/Downloads/openwrt-fanqiang
CONFIG=~/Downloads/openwrt-dir505

createdir() {
    rm -rf $CONFIG
    mkdir $CONFIG
}

copy() {
    cp -R $REPOSITORY/openwrt/default/* $CONFIG/
    cp -R $REPOSITORY/openwrt/dir505/* $CONFIG/
}

setmod() {
    chmod +x $CONFIG/usr/bin/ss-firewall-asia
    chmod +x $CONFIG/etc/uci-defaults
    chmod +x $CONFIG/etc/uci-defaults/*
}

modify() {
    # server ip address
    sed -i 's/1.0.9.8/server_ip/' $CONFIG/etc/shadowsocks-libev/config.json

    # server_port
    sed -i 's/1098/server_port/' $CONFIG/etc/shadowsocks-libev/config.json

    # local_port
    sed -i 's/7654/7654/' $CONFIG/etc/shadowsocks-libev/config.json

    # password
    sed -i 's/killfw/killfw/' $CONFIG/etc/shadowsocks-libev/config.json
```

```

# method
sed -i 's/chacha20-ietf-poly1305/chacha20-ietf-poly1305/' $CONFIG/etc/shadowsocks-libev/config.json

# server ip addresss
sed -i 's/1.0.9.8/server_ip/' $CONFIG/usr/bin/ss-firewall-asia

# local_port
sed -i 's/7654/7654/' $CONFIG/usr/bin/ss-firewall-asia

# ppoe username
sed -i 's/wan-username/wan-username/' $CONFIG/usr/bin/router

# ppoe password
sed -i 's/wan-password/wan-password/' $CONFIG/usr/bin/router

# wifi password
sed -i 's/icanfly9876/icanfly9876/g' $CONFIG/usr/bin/ap
sed -i 's/icanfly9876/icanfly9876/g' $CONFIG/usr/bin/router

# root password
sed -i 's/\\nfanqiang/\\nfanqiang/' $CONFIG/etc/uci-defaults/defaults
}

if [ "$1" = "createdir" ]; then
    createdir
elif [ "$1" = "copy" ]; then
    copy
elif [ "$1" = "setmod" ]; then
    setmod
elif [ "$1" = "modify" ]; then
    modify
else
    echo "usage: createdir copy setmod modify"
fi

```

用法: 在 config-dir505.sh所在目录运行:

```

./config-dir505.sh createdir
./config-dir505.sh copy
./config-dir505.sh setmod
./config-dir505.sh modify

```

开始编译OpenWrt自动翻墙固件

```

cd ~/Downloads/openwrt-imagebuilder
make image PROFILE=DIR505A1 PACKAGES="libiwinfo-lua liblua liblucihttp liblucihttp-lua libubus-lua lua luci luci-app-firewall luci-base luci-lib-ip
luci-lib-jsonc luci-lib-nixio luci-mod-admin-full luci-proto-ipv6 luci-proto-ppp luci-theme-bootstrap rpcd rpcd-mod-rrdns uhttpd base-files libc l
ibgcc busybox dropbear mtd uci opkg netifd fstools uclient-fetch logd kmod-gpio-button-hotplug swconfig kmod-ath9k wpad-mini uboot-envtools iptable
s ip6tables ppp ppp-mod-pppoe firewall odhcpd-ipv6only odhcp6c kmod-usb-core kmod-usb-ohci kmod-usb2 kmod-usb-ledtrig-usbport wget bind-dig iptable
s-mod-tproxy kmod-ipt-tproxy ip-full dnsmasq-full simple-obfs libmbdctl libcares libev libsodium shadowsocks-libev -dnsmasq" FILES=~/Downloads/ope
nwrtdir505

```

编译好的固件在ImageBuilder的bin/targets/ar71xx/目录下

然后把编译出的固件刷进路由器, 重启路由器后就能免设置智能翻墙

升级固件要用到的是 ...sysupgrade.bin, 如果在原厂固件上刷要用 ...-factory.bin

先本地修改好配置文件再编译, 然后把翻墙固件刷进D-Link DIR-505 A1后, 就能零设置智能、自动翻墙










只要配置文件设置不出差错, 编译固件一般都能成功, 保存好这个固件, 以后随便折腾路由器, 出现问题大不了重刷一次, 几分钟时间就一切都恢复正常

相关资源:

- <https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>
- <https://software-download.name/2014/shadowsocks-libev-polarssl-ar71xx-ipk-latest/>
- <https://openwrt.org/zh-cn/doc/howto/obtain.firmware.generate>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05

编译OpenWrt全自动翻墙固件 for D-Link DIR-505 A1

-  下载适合D-Link DIR505无线路由器的Image Builder
-  确定OpenWrt无线路由器的PROFILE值
-  找出默认应该包含进OpenWrt固件的包
-  按照教程 编译shadowsocks-libev for OpenWrt ipk安装包
-  下载和设定自定义翻墙配置文件
-  设置可执行权限
-  必须修改的DIR505翻墙配置文件:
-  自动复制和修改DIR-505翻墙设置文件
-  开始编译OpenWrt自动翻墙固件

D-Link DIR-505 A1 刷通用OpenWrt固件

照前面的教程自己编译翻墙固件，编译出来后刷进路由器，就能实现零设置自动翻墙。出于各种原因，有的朋友可能不想自己编译固件，又想用DIR-505实现智能翻墙，就要下载预编译的通用翻墙固件，刷好后，登录路由器，用vi修改少数几个设置，就能实现智能翻墙，本教程就是针对这些朋友而写

路由器的开关拨到刻有 Router/AP 字样的档位，如果你没有给路由器动过手术，就是从上往下数的第一档

DIR-505原厂固件刷翻墙固件的方法

适合购买了D-Link DIR-505 A1后没有刷过任何固件的朋友

刷修改了16进制值的原厂固件：

到下面地址下载已经修改了16进制值的原厂固件：<https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

照官方手册说明网页登录路由器，刷新固件

刷DIR-505的翻墙固件 **factory.bin**：

到下面地址下载用于 DIR-505的翻墙固件：

<https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

下载 openwrt-ar71xx-generic-dir-505-a1-squashfs-factory.bin

按照官方手册的说明刷新固件

OpenWrt固件基础上升级到翻墙固件

注意，下面步骤适合于你已经在你的DIR-505上刷了OpenWrt固件,你想要升级到可以自己翻墙的openwrt固件

下载翻墙固件 **sysupgrade.bin**：

到下面地址下载用于 DIR-505的翻墙固件 openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade.bin: <https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

命令行上传固件到路由器：

电脑通过网线或无线连接到路由器，然后：

```
cd ~/Downloads/openwrt-imagebuilder/bin/targets/ar71xx/
scp openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade.bin root@192.168.1.1:/tmp/
```

ssh登录OpenWrt路由器 ssh root@192.168.1.1 cd /tmp

sysupgrade升级固件并取消保留原来配置文件：

```
root@OpenWrt:/tmp# sysupgrade -n openwrt-ar71xx-generic-dir-505-a1-squashfs-sysupgrade.bin
```

参数 **-n** 表示升级时不保留原来的配置文件

等待两分钟等刷新固件并重启完成

相关资源：

- <https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-12-05

[D-Link DIR-505 A1 刷通用OpenWrt固件](#)

-  DIR-505原厂固件刷翻墙固件的方法
-  OpenWrt固件基础上升级到翻墙固件

登录并设置 DIR-505 OpenWrt 翻墙固件

ADSL Modem网线连接到路由器的有线接口。路由器的开关拨到刻有 Router/AP 字样的档位, 如果你没有给路由器动过手术, 就是从上往下数的第一档。本文以router模式为例, 如果你的应用场景是ap模式, 请自行相应变通

电脑连接DIR-505路由器

电脑连接到无线 网络 **eastking-dir505**

无线密码:

```
2014-09-01版: wsjdw,8181
新版都是: icanfly9876
```

ssh 登录 OpenWrt 翻墙固件

```
ssh root@192.168.1.1
```

输入密码 **fanqiang** 登录ssh

有时会提示错误:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@  WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!  @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
cf:c5:12:34:56:0b:4d:1c:56:48:6a:87:04:cf:b8:83.
Please contact your system administrator.
Add correct host key in /home/openwrt-fanqiang/.ssh/known_hosts to get rid of this message.
Offending RSA key in /home/openwrt-fanqiang/.ssh/known_hosts:3
  remove with: ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1
RSA host key for 192.168.1.1 has changed and you have requested strict checking.
Host key verification failed.
```

解决办法就是复制并运行提示中的清理命令:

```
ssh-keygen -f "/home/openwrt-fanqiang/.ssh/known_hosts" -R 192.168.1.1
```

然后就可以正常登录了

登录后用vi修改设置:

```
root@OpenWrt:~# vi /etc/shadowsocks-libev/config.json
root@OpenWrt:~# vi /usr/bin/router
#如果是ap模式
root@OpenWrt:~# vi /usr/bin/ap
root@OpenWrt:~# vi /usr/bin/ss-firewall-asia
```

分别修改以下值:

- shadowsocks.json中, server改成你的服务器实际IP
- router/ap中 wan-username 和 wan-password改成实际值
- ss-firewall中, 1.0.9.8必须改成你的服务器实际IP

如果你还改了其他默认值, 请自行修改相应文件。不建议修改其他默认值, 以提高一次成功率。熟悉以后, 建议修改shadowsock密码

执行以下命令使修改生效


```
root@OpenWrt:~# /etc/init.d/shadowsocks restart
root@OpenWrt:~# /etc/init.d/dnsmasq restart
root@OpenWrt:~# /etc/init.d/network restart

# 查看 dnsmasq ss-redir ss-tunnel是否在运行。翻墙出现故障的时候也要查看:
ps
```





2015-12-24测试router模式, 修改配置文件, 编译出固件, 刷进路由器, 然后不用再修改任何设置就可以翻墙
等待约两分钟, 就可以测试是否可以在网上畅行无阻了

相关资源:

- <https://software-download.name/2014/dlink-dir-505-openwrt-fanqiang/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-05

[登录并设置 DIR-505 OpenWrt 翻墙固件](#)

-  电脑连接DIR-505路由器
-  ssh 登录 OpenWrt 翻墙固件
-  登录后用vi修改设置:
-  执行以下命令使修改生效

其他翻墙软件、方案教程

本教程主要内容是 路由器刷 OpenWrt, 安装 shadowsocks翻墙。有时也要用一下其他翻墙软件

最简单的路由器刷**OpenWrt**翻墙方案:

- <https://github.com/softwaredownload/openwrt-fanqiang>

在线阅读**OpenWrt**路由器翻墙、科学上网器教程:

- <https://fanqiang.software-download.name>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-10-22

利用lantern 蓝灯实现浏览器自动翻墙教程

蓝灯运用了多种技术, 通过自有服务器或者运行lantern的用户转发流量实现浏览器全自动翻墙

😁 lantern蓝灯和 OpenWrt shadowsocks翻墙的区别

- 蓝灯主要是浏览器自动翻墙
- 路由器OpenWrt shadowsocks翻墙方案 是所有接入的设备都自动翻墙,可定制性更高

🐼 为什么选择 lantern 蓝灯翻墙

有很多的翻墙软件, 有闭源的, 也有开源的, 我们优先选择开源软件。闭源软件缺少外界监督, 不能保证没有问题

蓝灯就是优秀的开源翻墙软件。今天是2016-01-10, 在Github上已经 6516 Star, 2228 Fork, 开发很活跃

😁 下载 lantern蓝灯翻墙软件

Github下载:

<https://github.com/getlantern/lantern>

主页下载:

<https://getlantern.org/>

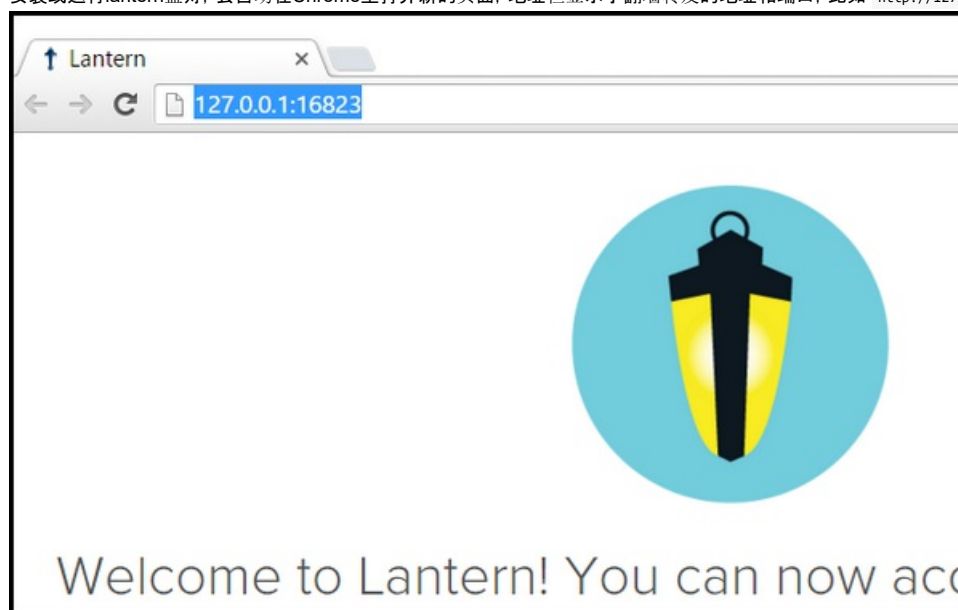
🐼 蓝灯翻墙软件安装和设置

- 停止路由器的shadowsocks翻墙
登录OpenWrt路由器, 运行命令:

```
/etc/init.d/shadowsocks stop
```

如果你是按照 <https://github.com/softwaredownload/openwrt-fanqiang> 设置的翻墙, 那么还得检查一下 [/etc/init.d/shadowsocks](#) 里的start, stop函数是否正确。2016-01-10前这两个函数有bug, 导致执行stop后上网不正常

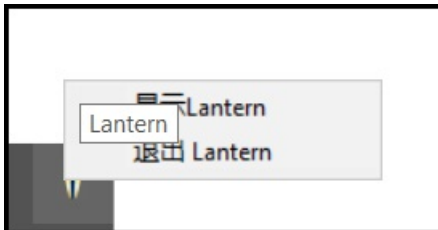
- 打开 [Chrome浏览器](#)
- 安装或运行lantern蓝灯, 会自动在Chrome里打开新的页面, 地址栏显示了翻墙转发的地址和端口, 比如 `http://127.0.0.1:16823/`



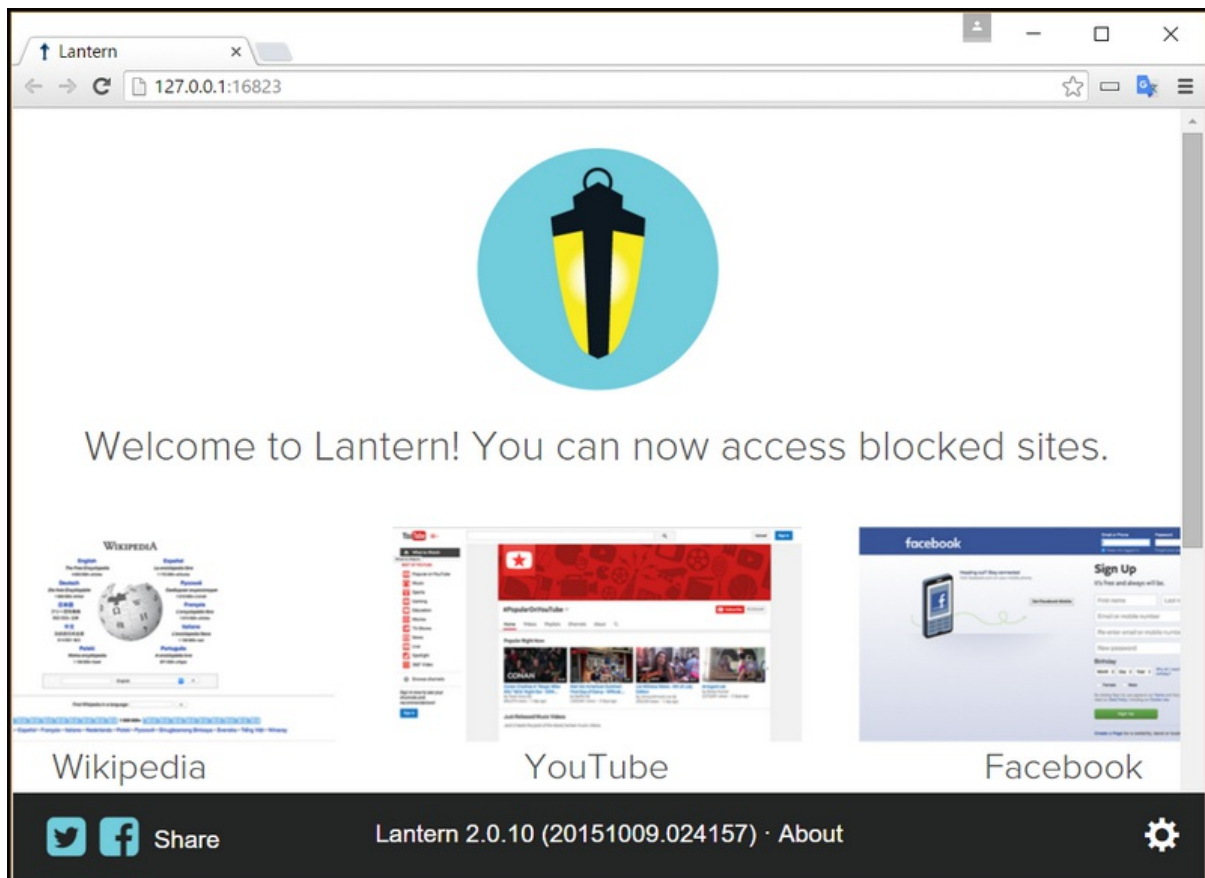
- 点击打开页面的右下角的齿轮图标设置lantern翻墙配置:



- 右键点击电脑右下角托盘图标退出lantern(Windows 为例)



如果一切正常，一运行蓝灯，就可以点击蓝灯新打开的页面上的 YouTube 图标看视频了，非常方便



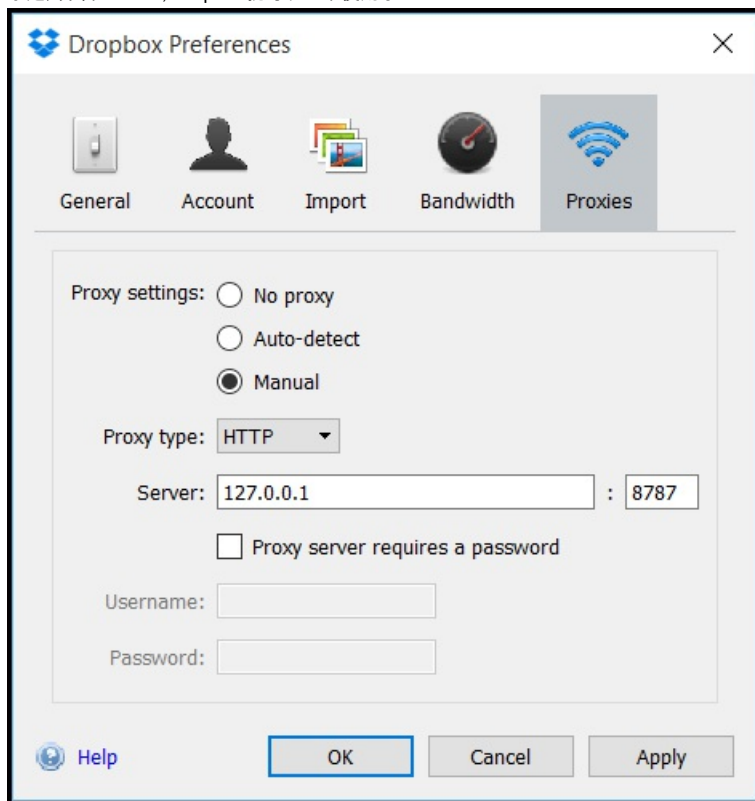
🏠 配置网络软件走 **Lantern** 翻墙代理:

蓝灯默认会在 127.0.0.1 上开启一个 HTTP 代理,端口号是 8787
在网络软件的代理界面上设置 HTTP 代理:

地址: 127.0.0.1
端口号: 8787

(注:“127.0.0.1”表示“本机地址”)

于是, 开启Lantern, Dropbox就可以正常使用了:



lantern蓝灯翻墙软件配置文件研究

进入lantern蓝灯翻墙软件安装目录:

Windows下进入lantern安装目录:

按Windows键, 输入
%appdata%

然后就可以进入 Lantern 安装目录

Mac 下进入lantern安装目录:

```
/Users/name/Library/Application Support/Lantern
```

配置文件:**Lantern/lantern-2.0.10.yaml**:

2.0.10是版本号, 随不同版本而变化

log 文件, 可以了解翻墙详细过程:

```
Lantern/logs/lantern.log
...
geolookup.go:161 Successfully looked up IP '1.0.9.8' and country 'CN'
...
```

Lantern配置文件中的流量转发服务器IP地址:

Lantern/lantern-2.0.10.yaml 中找到类似如下内容, 替换成其他服务器, 把文件设为只读, 就可以更换服务器:

```
fallback-1.0.9.8:
  addr: 1.0.9.8:443
  pipelined: false
  cert: "-----BEGIN CERTIFICATE-----\n...\n-----END
  CERTIFICATE-----\n"
  authtoken: B... https://github.com/softwaredownload/openwrt-fanqiang ...C
```

Ubuntu下自己编译lantern翻墙软件:

先准备好Go语言开发环境, 假设Go程序的源码在 `~/golib/src` 目录下

```
sudo apt-get update
sudo apt-get install -y git curl libappindicator3-dev build-essential libgtk-3-dev

# Use the Go compiler to build the lantern binary
cd ~/golib/src
git clone https://github.com/getlantern/lantern.git

cd lantern
source setenv.bash
go build -o lantern github.com/getlantern/flashlight

# Use curl to test that the proxy is working fine:
curl -x 127.0.0.1:8787 https://www.google.com/humans.txt

# This line will run Lantern without opening the browser window:
./lantern -headless
```

相关资源:

- <https://github.com/getlantern/lantern>
- <https://getlantern.org>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

利用lantern 蓝灯实现浏览器自动翻墙教程

-  lantern蓝灯和 OpenWrt shadowsocks翻墙的区别
-  为什么选择 lantern 蓝灯翻墙
-  下载 lantern蓝灯翻墙软件
-  蓝灯翻墙软件安装和设置
-  配置网络软件走 Lantern 翻墙代理:
-  lantern蓝灯翻墙软件配置文件研究

怎样加强上网的匿名性

即使翻墙上网了, 真实的上网信息, 如本机IP地址, 系统语言, 系统时区等等还是可能暴露

🚗 怎样检查翻墙后浏览器上网的匿名性

访问下面网站检查自己的匿名程度:

<https://whoer.net/#extended>

💡 蓝灯翻墙, 浏览器匿名程度测试

下图, 蓝灯翻墙, Chrome浏览器, 匿名程度 40%, 很差:

The screenshot shows the 'My IP' section with a redacted IP address. The 'Your anonymity' score is 40%, with a warning 'Too much is known about you!'. The 'Location' section shows the user is in China. The 'Browser' section shows 'Win10.0' and 'Chrome 47.0'. The 'DNS' section shows a redacted IP address. The 'Proxy' section shows 'No'. The 'TOR' section shows 'No'. The 'Anonymizer' section shows 'No'. The 'Blacklist' section shows 'No'.

再拉下去看, WebRTC暴露了本机IP地址:

The screenshot shows the 'Interactive detection' section with a 'Run tests' button. The 'IP address' section shows 'github.com/softwaredownload/openwrt-fanqiang'. The 'WebRTC' section shows a redacted IP address and 'China'. The 'Location' section shows 'Country: Singapore', 'Continent: Asia', 'Region: N/A', 'City: Singapore', and 'ZIP: N/A'.

下图, 蓝灯翻墙, FireFox浏览器, 开启隐私设置后WebRTC已经关闭, 匿名程度高达90%:

The screenshot shows the 'My IP' section with a redacted IP address. The 'Your anonymity' score is 90%, with a note 'Minor remarks regarding your anonymity and security'. The 'Location' section shows the user is in Singapore. The 'Browser' section shows 'Win10.0' and 'Firefox 43.0'. The 'DNS' section shows a redacted IP address. The 'Proxy' section shows 'No'. The 'TOR' section shows 'No'. The 'Anonymizer' section shows 'No'. The 'Blacklist' section shows 'No'.

😄 路由器刷OpenWrt, 安装shadowsocks-libev翻墙, 浏览器匿名程度测试

下图, FireFox浏览器, 同样设置, WeRTC已经关闭, 匿名程度64%:

The screenshot shows a web-based IP leak test interface. At the top, it displays 'My IP:' followed by a redacted IP address. To the right, a green banner indicates 'Your anonymity: 64%' with the text 'Serious security and anonymity fails'. Below this, a green box contains the URL 'https://github.com/softwaredownload/openwrt-fanqiang'. The interface is divided into two columns. The left column lists system information: Location (USA), ISP (Digital Ocean), Hostname (N/A), OS (Win10.0), and Browser (Firefox 43.0). The right column lists security features: Proxy (No), TOR (No), Anonymizer (No), and Blacklist (No), each with a thumbs-up icon.

Chrome浏览器, 匿名程度只有30%了:

The screenshot shows the same web-based IP leak test interface but for Chrome. The 'Your anonymity' banner is red and shows '30%' with the text 'Too much is known about you!'. The URL box is empty. The system information on the left is the same, but the browser is listed as 'Chrome 47.0'. The security features on the right are the same, but the DNS field is now visible and redacted, and the TOR field is also redacted.

🚫 防止浏览器 WebRTC 泄露本机IP地址

Chrome浏览器安装插件就可以了: WebRTC Leak Prevent:

安装以后, 路由器刷OpenWrt, 安装shadowsocks-libev翻墙, Chrome浏览器的匿名程度提升到了64%

Firefox浏览器关闭 WebRTC:

地址栏输入: about:config

搜索: media.peerconnection.enabled 双击由true改为false, 就可以彻底匿名了!

Opera浏览器安装插件: WebRTC Leak Prevent:

什么是WebRTC What is WebRTC:

WebRTC, 名称源自网页实时通信(Web Real-Time Communication)的缩写, 是一个支持网页浏览器进行实时语音对话或视频对话的技术, 是谷歌2010年以6820万美元收购Global IP Solutions公司而获得的一项技术

WebRTC实现了基于网页的视频会议, 标准是WHATWG 协议, 目的是通过浏览器提供简单的javascript就可以达到实时通讯(Real-Time Communications (RTC))能力

WebRTC(Web Real-Time Communication)项目的最终目的主要是让Web开发者能够基于浏览器(Chrome\FireFox...)轻易快捷开发出丰富的实时多媒体应用, 而无需下载安装任何插件, Web开发者也无需关注多媒体的数字信号处理过程, 只需编写简单的Javascript程序即可实现, W3C等组织正在制定Javascript 标准API, 目前是WebRTC 1.0版本, Draft状态; 另外WebRTC还希望能够建立一个多互联网浏览器间健壮的实时通信的平台, 形成开发者与浏览器厂商良好的生态环境。同时, Google也希望和致力于让WebRTC的技术成为HTML5标准之一, 可见Google布局之深远 WebRTC提供了视频会议的核心技术, 包括音视频的采集、编解码、网络传输、显示等功能, 并且还支持跨平台: windows, linux, mac, android

相关资源:

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

怎样加强上网的匿名性

- 🌐 怎样检查翻墙后浏览器上网的匿名性
- 🚦 蓝灯翻墙, 浏览器匿名程度测试
- 🏠 路由器刷OpenWrt, 安装shadowsocks-libev翻墙, 浏览器匿名程度测试
- 🚫 防止浏览器 WebRTC 泄露本机IP地址

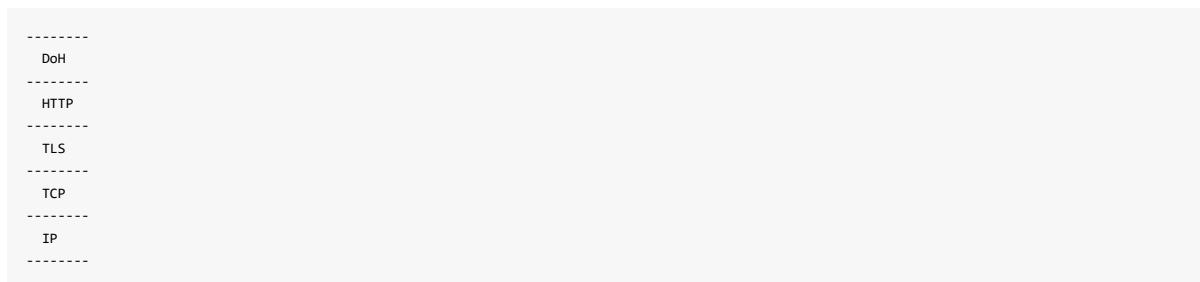
配置浏览器使用 DNS over HTTPS (DoH) 进行安全 DNS

什么是 DNS over HTTPS

域名安全协议有如DNSSEC, DNSCrypt, DNS over TLS, DNS over HTTPS, 而 DNS over HTTPS 最被看好

DNS over HTTPS 简称为 **DoH** 是基于 HTTPS 隧道之上的域名协议。HTTPS 流量特征目前无法识别, 那么 DoH 也就无法识别, 白脸不知道你是不是在浏览 https 网站还是在进行 DNS 查询, 所以很安全

DoH 协议栈示意



DNS over HTTPS 缺点

相比DNS over TLS (DoT), DoH 多了一层封装, 所以性能会比 DoT 略差, 如果使用国内的DoH服务, 这个性能损失是可以忽略的

为什么推荐使用 DNS over HTTPS

- 基于 HTTPS 之上, 十分安全。白脸不知道你在进行域名查询
- 基于 HTTPS 之上, 可以无缝支持 Proxy
- 可以充分利用 HTTP 2.0 的特性
- 浏览器积极支持

Firefox 从 63.0 beta 开始正式支持 DoH

本教程使用 DoH 的环境

- 路由器配置好了 shadowsocks 翻墙服务
 - shadowsocks-libev 客户端 ss-redir 提供流量翻墙
 - dnsmasq 分配 dns 查询
 - shadowsocks-libev 客户端 ss-tunnel 转发 DNS 查询到 shadowsocks 服务端
- 电脑或其他设备的网络连接属性中, 网关和DNS设为路由器地址

此时所有连上路由器的设备都可以自动翻墙

浏览器设置为使用国内DoH服务端进行DNS解析, 也就是浏览器直接进行DNS查询, 不通过路由器 dnsmasq 和 ss-tunnel 进行转发了

这样做的好处是减轻了路由器的负担, 并且DNS查询的速度可能比转发到 shadowsocks 服务端更快

如果只有浏览器需要用到翻墙服务, 那么所有浏览器都配置 DoH, 就可以把路由器里的 dnsmasq 和 ss-tunnel 停掉, 同时网络连接属性中的 DNS 没有必要设为路由器地址了

浏览器使用DNS over HTTPS (DoH)的准备工作

我们要使用国内的 DoH 服务端, 需要先把 DoH服务端的域名和 IP 地址加入到路由器的相应配置中

- DoH 服务端域名加入到路由器 dnsmasq 国内网站名单中
- DoH 服务端IP地址加入到路由器防火墙的忽略列表中

如果你按照 [OpenWrt 路由器 shadowsocks自动翻墙](#)、[科学上网教程](#)

<https://github.com/softwaredownload/openwrt-fanqiang>

配置了路由器自动翻墙，那么就很简单了，步骤如下：

- 把项目 clone 到本地，假定是 C 盘根目录

```
git clone https://github.com/softwaredownload/openwrt-fanqiang.git
```

- 把相关文件复制到路由器，假设你使用的是 Git Bash for Windows

```
cd /C/openwrt-fanqiang
scp openwrt/default/etc/dnsmasq.d/custom.china.conf root@192.168.1.1:/etc/dnsmasq.d/
```

`custom.china.conf` 是自定义的在国内进行 dns 的域名，已经把我们要用到的 DoH 服务端域名加入其中了

```
scp openwrt/default/etc/shadowsocks-libev/ip_custom.txt root@192.168.1.1:/etc/shadowsocks-libev/
```

`ip_custom.txt` 是自定义的防火墙规则中需要忽略的IP，已经包含了 DoH 服务端的 IP 地址

我们把数据从防火墙设置脚本中分离了出来，改动数据不需要去动脚本文件，十分方便

需要注意的是，`ip_custom.txt` 等数据文件不能使用 Windows 记事本编辑，可以使用第三方编辑器如 Sublime Text，并把换行方式设置为 Linux 格式

- 登录路由器，执行命令使用新数据生效

```
ssh root@192.168.1.1
kige@openwrt:~# /etc/init.d/dnsmasq restart
kige@openwrt:~# /etc/init.d/shadowsocks restart
```

Firefox 配置使用 DNS over HTTPS (DoH)

- 下载 Firefox

Firefox 自从 63.0 版本开始，提供了十分简单的 DoH 配置界面

如果你使用的是 63.0 以前的版本，先卸载它

Firefox配置DoH方法参考这个[教程](#)

- 测试浏览器 DoH 是否起作用

打开一些外网，如 <https://youtube.com> <https://flickr.com>

Firefox地址栏输入 `about:networking#dns` 查看有哪些域名是通过 DoH 服务解析的

TRR = Trusted Recursive Resolver，结果中 TRR 列为 true 表示域名是通过 DoH 解析的

也可以路由器关闭 dnsmasq再测试：

```
kige@openwrt:~# /etc/init.d/dnsmasq stop
```

这时别的浏览器没有配置过DoH，又无法通过路由器解析域名，自然打开 youtube.com，只有Firefox还是畅行国内外无阻

目前用的是红鱼DNS，可能是技术原因，有的网站可能无法解析，切换到未用 DoH 的浏览器就正常了

相关资源：

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/default/etc/dnsmasq.d>
- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/openwrt/default/etc/shadowsocks-libev>
- <https://www.rubyfish.cn/config-firefox>
- <https://fanqiang.software-download.name/>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-12-07

[配置浏览器使用 DNS over HTTPS \(DoH\) 进行安全 DNS](#)

-  什么是 DNS over HTTPS
-  DoH 协议栈示意
-  DNS over HTTPS 缺点

-  为什么推荐使用 DNS over HTTPS
-  本教程使用 DoH 的环境
-  浏览器使用DNS over HTTPS (DoH)的准备工作
-  FireFox 配置使用 DNS over HTTPS (DoH)

全面优化翻墙系统

经过测试, 翻墙系统经过优化以后, 可以显著提高翻墙上网的速度, 使用 [Digital Ocean](#) New York 数据中心的 VPS, youtube.com 1080P 视频无压力

一般情况下, 我们优化以下几项就可以了:

- 开启 TCp fast open
- 开启 BBR 加速
- 优化打开文件数目
- 设置 swap 交换文件

最简单的路由器刷OpenWrt翻墙方案:

- <https://github.com/softwaredownload/openwrt-fanqiang>

在线阅读OpenWrt路由器翻墙、科学上网教程:

- <https://fanqiang.software-download.name>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-10-22

Ubuntu OpenWrt 开启 TCP Fast Open (TFO)流量加速

🤖 什么是 TCP Fast Open - TFO

TCP Fast Open, 简称TFO, 意思是TCP快速打开, 是对计算机网络中传输控制协议(TCP)连接的一种简化握手手续的拓展, 用于提高两端点间连接的打开速度

二个人只有在认识的基础上才可能深入交流。客户端和服务要传递数据, 也需要先“认识”, 称为“握手”, 握手成功才正式开始传送数据

服务端和客户端传递数据前需要握手三次, 这会导致延时, 启用 TFO 后, 如果验证成功, 它可以在三次握手最终的ACK包收到之前就开始发送数据, 这样便跳过了一个绕路的行为, 于是在传输开始时就降低了延迟

也就是说, TFO 降低了握手阶段的延迟, 至于握手成功后数据传递的速度, 和 TFO 是没有关系的

🔧 开启 TFO 的先决条件

Linux kernel 3.7 及以上才支持 TCP Fast Open

在服务端的Ubuntu 检查一下:

```
uname -r
4.15.0-34-generic
```

再登录客户端 OpenWrt 路由器确认一下是否可以开启 TCP Fast Open

```
uname -r
4.9.120
```

Linux kernel 3.13 及以上默认已经开启了TFO, Linux服务器上验证方法:

```
cat /proc/sys/net/ipv4/tcp_fastopen
1
```

如果返回 0 表示没有开启TFO, 非0则是默认开启了

🐼 tcp_fastopen三个数值选项的含义

tcp_fastopen选项是二进制位掩码, 其中第一位启用或禁用客户端支持(默认开启), 第二位设置服务端支持(默认关闭), 第3位设置是否允许SYN数据包中的数据而不使用TFO cookie选项

- tcp_fastopen = 1
只能在传出连接上启用(仅限客户端)
- tcp_fastopen = 2
仅在侦听套接字(服务端)上允许TFO
- tcp_fastopen = 3
客户端和服务端都启用TFO

请注意, 即使启用了这些选项, 也必须启用应用程序级支持。我们在服务端和客户端的操作系统上启用了 tcp fast open, 如果软件层面不支持, 还是起不到TCP通信加速的作用

😄 临时启用双向 tcp_fastopen

```
$ sudo sysctl -w net.ipv4.tcp_fastopen=3
$ cat net.ipv4.tcp_fastopen
3
```

查看一下 sysctl -w的用法:

```
$ sysctl --help | grep write
-w, --write          enable writing a value to variable
```

-w 意思是把指定值写入变量

Ubuntu系统开启 TFO 加速

通过 sysctl 我们可以定义内核参数

自定义 TFO 有两种途径:

- /etc/sysctl.conf
- /etc/sysctl.d/*.conf

这两种途径有什么区别?

/etc/sysctl.d/README:

This directory contains settings similar to those found in /etc/sysctl.conf. In general, files in the 10-*.conf range come from the *procps* package and serve as system defaults. Other packages install their files in the 30-*.conf range, to override system defaults. End-users can use 60-*.conf and above, or use /etc/sysctl.conf directly, which overrides anything in this directory.

/etc/sysctl.d/目录包含与/etc/sysctl.conf中类似的设置。通常, 10-*.conf范围内的文件来自*procps*包和 作为系统默认值。其他包安装他们的文件 30-*.conf范围, 覆盖系统默认值。最终用户可以使用60-*.conf 以上, 或直接使用/etc/sysctl.conf, 它会覆盖任何内容 这个目录

也就是开头数字小的.conf先加载, 后面的覆盖前面, 最后加载/etc/sysctl.conf

网上的教程一般是把 TFO 设置写在 /etc/sysctl.conf, 那么哪一种方式最佳呢

有少数的软件包建议直接编辑/etc/sysctl.conf, 但这可能是为了与旧的GNU / Linux发行版的兼容, 最近软件包一般建议使用.d配置目录, 这样更加灵活

用的.d目录(如/etc/sysctl.d/)的重点是允许应用程序/管理员在那里添加文件, 这比添加或删除单个文件中的条目更容易, 更安全。例如, 可以将 TFO的参数放在文件/etc/sysctl.d/98-tcp_fastopen.conf中, 一看就知道这些参数是针对tcp_fastopen的。任何自动化工具(包括安装程序)也更容易将文件放在目录中而不是附加到现有文件

明白了原理以后, 自然尽量不去编辑 /etc/sysctl.conf 了

```
$ su
# echo 'net.ipv4.tcp_fastopen=3' > /etc/sysctl.d/98-tcp_fastopen.conf
# reboot
$ cat /proc/sys/net/ipv4/tcp_fastopen
3
```

上面命令, 我们切换到 root 用户, 把设置写入文件, 然后重启Ubuntu验证, 返回3说明设置正确

服务端生成 TCp fast open 持久化密钥

Linux 内核3.13开始, 在应用程序首次设置相关的setsockopt系统调用选项时生成密钥。在设置密钥之前, proc值全为零

默认情况下, 由于密钥在系统重新启动之后不会持续存在, 因此正规地使用TFO, 应该包括通过sysctl安全地保存密钥, 比如生成随机密钥, 设置限制性文件权限。这将确保客户端可以使用现有cookie而无需生成新密钥

要生成新密钥并通过sysctl进行持久化, 可以执行以下命令:

```
$ su
# RAND=$(openssl rand -hex 16)
# NEWKEY=${RAND:0:8}-${RAND:8:8}-${RAND:16:8}-${RAND:24:8}
# echo "net.ipv4.tcp_fastopen_key=$NEWKEY" > /etc/sysctl.d/98-tcp_fastopen_key.conf
# chmod 600 /etc/sysctl.d/98-tcp_fastopen_key.conf; chown root /etc/sysctl.d/98-tcp_fastopen_key.conf
# sysctl -p /etc/sysctl.d/98-tcp_fastopen_key.conf
# unset RAND NEWKEY
```

转换 TCP fast open 密钥

在Linux环境下, 有二种方法可以显示当TFO密钥:

- \$ sudo cat/proc/sys/net/ipv4/tcp_fastopen_key

- `$ sudo sysctl net.ipv4.tcp_fastopen_key`

同理，以可以用之改变密钥

TFO密钥是16个字节，表示为32个字符的十六进制字符串，分为4个8个字符的块，用短划线分隔，类似这样：

```
32100e0a-9876daaf-7654b836-21096051
```

可以使用前述方法实现密钥的转换

在多服务器服务器环境中，您需要随机生成一次密钥，并在所有服务器上设置相同的密钥

如果有多个翻墙VPS，需要随机切换使用，建议所有VPS使用相同密钥

OpenWrt系统开启 TFO 流量加速

```
vi /etc/sysctl.conf

# add line
net.ipv4.tcp_fastopen = 3
```

执行如下命令使之生效：

```
sysctl -p
```

Shadowsocks 软件启用 TCP Fast Open 加速

在服务端和 OpenWrt 路由器的 shadowsocks-libev 配置文件 config.json / shadowsocks.json 中加上：

```
"fast_open": true
```

重启服务端和客户端 shadowsocks-libev

测试 shadowsocks 服务端 TFO 有没有正常工作

登录shadowsocks服务端 Ubuntu 后，运行命令测试：

```
# 清空 IP 记录
sudo ip tcp_metrics flush all
ip tcp_metrics
# 应该是空
```

这时你打开 <https://www.youtube.com>

然后运行命令：

```
ip tcp_metrics
```

如果正常，会看到数条类似下面的记录

```
216.58.200.227 age 5034.824sec cwnd 10 rtt 185813us rttvar 185813us fo_mss 1380 fo_cookie 87c5a043c180c8ab source 210.98.76
216.58.200.227 age 5.356sec fo_mss 1380 fo_cookie 87aaa1f0c8761336 source 210.98.76
```

发现了 `fo_mss` 和 `fo_cookie` 字段没有，这是 TCP Fast Open 特有的字段，其中 `fo_cookie` 确认我们与行开头的IP地址216.58.200.227 的通信使用了TFO

如果系统还提供除了翻墙的其他网络服务，那么ip记录可能很多，其中一些记录并没有 `fo_cookie` 字段，可以运行命令把含有 `fo_cookie`的条目找出来：

```
ip tcp_metrics | grep fo_cookie
```

如果一条 `fo_cookie` 记录也找不到，说明开启 TFO 没有成功

🔧 查看 TCP fast open 的详细工作状态

root用户用这条命令查看TCP fast open的详细状态：

```
root@ubuntu:~# grep '^TcpExt:' /proc/net/netstat | cut -d ' ' -f 87-92 | column -t
```

我得到的值是：

TCPFastOpenMerge	TCPChallengeACK	TCP SYNChallenge	TCPFastOpenActive	TCPFastOpenActiveFail	TCPFastOpenPassive	TCPFastOpenPassiveFail	TCPFastOpenListenOverflow	TCPFastOpenCookieReqd	TCPFastOpenBlackhole	TCPSpuriousRtxHostQueues	BusyPollRxPackets	TCPAutoCorking	TCPFromZeroWindowAdv	TCPToZeroWindowAdv
1	12	5	79	23	621	1	70							
	2	1		0	0	0	48							48
	84													

如果非 root 用户执行，结果是：

TCPFastOpenPassiveFail	TCPFastOpenListenOverflow	TCPFastOpenCookieReqd	TCPFastOpenBlackhole	TCPSpuriousRtxHostQueues	BusyPollRxPackets
1	70	2	1	0	0

一定要注意，root用户才能查看 TCPFastOpenPassive 的值

🐱 测试 shadowsocks-libev 客户端有没有启用 TFO 加速

在shadowsocks-libev客户端 ss-rrdir | ss-tunnel 的启动参数中加上 -v 会在控制台显示 TFO 是否启用

```
kige@openwrt:~# /etc/init.d/shadowsocks stop
/usr/bin/ss-redir -v -b 0.0.0.0 -c /etc/shadowsocks-libev/config.json -f /var/run/shadowsocks.pid
```

☕ TCP fast open 状态值含义

- TCPFastOpenActive - 成功的出站TFO连接数
- TCPFastOpenActiveFail - 收到的SYN-ACK数据包的数量，这些数据包未确认SYN数据包中发送的数据并导致无SYN数据的重传。请注意原始SYN数据包包含cookie + 数据，这不是连接到不支持TFO的服务器的数量
- TCPFastOpenPassive - 成功的进站TFO连接数
- TCPFastOpenPassiveFail - TFO cookie无效的进站SYN数据包数
- TCPFastOpenCookieReqd - 请求TFO设置但没有cookie的进站SYN数据包数
- TCPFastOpenListenOverflow - 由于套接字已超过最大队列长度而将禁用TFO的进站SYN数据包的数量

上面检测 TCP fast open 状态，得到：

```
TCPFastOpenPassive
621
```

成功的进站TFO连接数621，这是各项值中的最大值，说明 TFO 工作正常

🐛 常见问题

• TFO丢包

当带有TFO的包经过路由器 可能会被丢包 不同运营商的策略也不同 如果配置了NAT地址池 在第二次连接时可能会成功 取决于NAT表老化时间 客户端IP改变和NAT之后的公网IP改变都会影响TFO的正常使用 服务端收到不正确的包后依旧可以发送syn-ack 且退回为3WHS 可以选择在服务端单方面将模式改为1或者客户端改为0 不影响服务端对外连接的性能

相关资源：

- https://github.com/softwaredownload/openwrt-fanqiang/blob/master/ubuntu/etc/sysctl.d/98-tcp_fastopen.conf
- <https://bradleyf.id.au/nix/shaving-your-rtt-wth-tfo/>
- [维基 TCP快速打开](#)
- https://wikitech.wikimedia.org/wiki/TCP_Fast_Open
- <https://www.keycdn.com/support/tcp-fast-open/>
- [8.6 TCP Fast Open\(TFO\)](#)

- 各种加密代理协议的简单对比
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name>

2018-12-07

Ubuntu OpenWrt 开启 TCP Fast Open (TFO)流量加速

-  什么是 TCP Fast Open - TFO
-  开启 TFO 的先决条件
-  tcp_fastopen三个数值选项的含义
-  临时启用双向 tcp_fastopen
-  Ubuntu系统开启 TFO 加速
-  服务端生成 TCp fast open 持久化密钥
-  转换 TCP fast open 密钥
-  OpenWrt系统开启 TFO 流量加速
-  Shadowsocks 软件启用 TCP Fast Open 加速
-  测试 shadowsocks 服务端 TFO 有没有正常工作
-  查看 TCP fast open的详细工作状态
-  测试 shadowsocks-libev 客户端有没有启用 TFO 加速
-  TCP fast open 状态值含义
-  常见问题

Shadowsocks 服务端 Ubuntu 开启BBR加速

关于 BBR 加速算法

BBR是一款Google开发的TCP拥塞控制算法, 开启这个算法的好处:

- 在有一定丢包率的网络链路上充分利用带宽。非常适合高延迟, 高带宽的网络链路
- 降低网络链路上的buffer占用率, 从而降低延迟。非常适合慢速接入网络的用户

先检查一下 Ubuntu 系统是否可以开启这个加速算法

```
uname -r
4.15.0-36-generic
```

Linux系统内核高于 4.9 就可以开启。如果你的系统内核低于4.9, 升级 Ubuntu 到最新版本就可以了

下面默认 Ubuntu 内核版本高于4.9, 基于KVM的 VPS(包括DO)

shadowsocks 服务端开启 BBR加速

开启 BBR 加速需要设置 Linux 内核参数。自定义内核参数最好的实践是这样的:

- 在 /etc/sysctl.d/ 下设置, 尽量避免修改 /etc/sysctl.conf
- 文件名以数字开头, 表示系统启动时文件加载的顺序, 数字小的文件先加载, 最后加载 sysctl.conf
- 文件名应该表示明确的意义, 比如 50-tcp_fastopen.conf

```
kige@ubuntu:~$ su
# echo 'net.core.default_qdisc=fq' > /etc/sysctl.d/98-bbr.conf
# echo 'net.ipv4.tcp_congestion_control=bbr' >> /etc/sysctl.d/98-bbr.conf
# sysctl --system
```

上面命令切换到 root 用户, 把设置写入 .conf, 然后用 sysctl --system 从系统目录重新读入所有配置

不要用 sysctl -p 来代替 sysctl --system, 因为 sysctl --system 和重启系统的效果类似, 于是我们可以测试出重启系统后的情况

用了 98-bbr.conf 而不是 50-bbr.conf, 是为了覆盖 50-default.conf 中设定的默认值: net.core.default_qdisc = fq_codel

fq 和 fq_codel 有什么区别

CoDel 是 controlled delay的缩写

- net.core.default_qdisc = fq_codel

最好的通用qdisc

- net.core.default_qdisc = fq

用于胖服务器, fq_codel用于路由器, 在虚拟化环境中, 底层服务器是路由器, 客户虚拟机是主机

检查 BBR 模块有没有启动:

```
kige@ubuntu:~$ lsmod | grep bbr
```

返回值有 tcp_bbr 说明 bbr 已启动

再检查我们刚才设置的值是否已经起作用:

```
$ sysctl net.ipv4.tcp_congestion_control
net.ipv4.tcp_congestion_control = bbr
$ sysctl net.core.default_qdisc
net.core.default_qdisc = fq
```

相关资源:

- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/ubuntu/etc/sysctl.d/98-bbr.conf>
- <https://software-download.name/2014/fanqiang-jiaocheng/>

- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name>
Shadowsocks 服务端 Ubuntu 开启BBR加速

2018-12-07

-  关于 BBR 加速算法
-  shadowsocks 服务端开启 BBR加速

Ubuntu server 最大打开文件数目优化

如何查看系统打开文件数量的限制

如何在 Linux 系统下增加打开文件的最大数量？如何在Linux下打开更多文件描述符？我们要根据系统原先的设定来决定是否要调整设置

可以使用以下命令显示系统允许的最大打开文件描述符数：

```
$ cat /proc/sys/fs/file-max
97898
```

我在 Ubuntu 18.04 上得到的最大打开文件描述符数量是 97898, 也就是正常用户可以在单个登录会话中打开97898个文件

先查看一下当前用户会话的打开文件数的软限制和硬限制, 可以使用下面命令：

```
$ ulimit -Sn
1024
$ ulimit -Hn
1048576
```

命令选项中 S 即 soft, H 即 hard

如何调整系统范围的文件描述符(FD)限制

可以通过Linux操作系统下的/etc/sysctl.conf文件更改整个系统中同时打开的文件描述符的数量

有时系统可能发生达到最大文件数的错误, 如何解决此问题？

许多应用程序(如Oracle数据库或Apache Web服务器)需要此范围相当高。因此, 您可以通过在内核变量 /proc/sys/fs/file-max 中设置新值来增加打开文件的最大数量

前面上面我们查看过, 系统打开文件数限制是 97898, 这个数字一般来说够大了, 如果需要修改, 可以如下设置

如下所示(以root身份登录)：

```
#sysctl -w fs.file-max = 51200
```

以上命令强制限制为 51200 个文件, 并即时生效

您需要编辑/etc/sysctl.conf文件并放入以下行, 以便重新启动后设置仍然生效：

```
# vi /etc/sysctl.conf
fs.file-max = 51200
```

需要注意的是, 直接修改 /etc/sysctl.conf 给管理带来了不便, 最好是把所有的自定义设置放在单独文件中, 这样我们备份系统或测试时就很安全、方便了

因此, 我们应该在 /etc/sysctl.d/ 目录下创建一个 [98-file-max.conf](#) 文件, 把自定义设置写在那里

```
# vi /etc/sysctl.d/98-file-max.conf
fs.file-max = 51200
```

保存并关闭文件。用户需要注销并重新登录才能使更改生效, 或者只需键入以下命令：

```
# sysctl --system
```

使用命令验证您的设置：

```
# cat /proc/sys/fs/file-max
51200
```

或者这样：

```
# sysctl fs.file-max
51200
```

☞ 如何设置用户级别打开文件描述符 FD 限制

上述过程设置了系统范围的文件描述符(FD)限制。但是,您可以通过编辑 `/etc/security/limits.conf` 文件 设置任意用户打开文件的限制

- 不同用户有单独的限制, 因此请确保以您的进程使用的用户身份运行此限制
- 有一个硬限制(hard), 一个软限制(soft)。soft是您的进程必须遵守的实际限制, hard设置了软限制可以设置的最大数量

从上面查看可知, 硬限制已经设置得比较大了, 我们需要修改的是软限制

请记住, ulimit 查看得到的数值不能保证你的进程实际拥有的限制! 在初始化shell之后(或之前), 有98种情况可以修改进程的限制数值

下面我们就来看一下 shadowsocks-libev 进程的限制数值

先查找一下 ss-server的进程 ID, 即PID

```
$ ps ax | grep ss-server
6543 ?        Ss          0:02 /usr/bin/ss-server -c /etc/shadowsocks-libev/config.json -u
```

得到进程ID后, 就可以这样查看 ss-server 的限制数值了

```
$ cat /proc/6543/limits
Limit                Soft Limit             Hard Limit              Units
Max cpu time          unlimited               unlimited               seconds
Max file size          unlimited               unlimited               bytes
Max data size          unlimited               unlimited               bytes
Max stack size         8388608                unlimited               bytes
Max core file size     0                      unlimited               bytes
Max resident set       unlimited               unlimited               bytes
Max processes          3841                   3841                    processes
Max open files         32768                  32768                   files
Max locked memory      16777216                unlimited               bytes
Max address space      unlimited               unlimited               bytes
Max file locks         unlimited               unlimited               locks
Max pending signals    3841                   3841                    signals
Max msgqueue size      819200                 819200                  bytes
Max nice priority      0                      0
Max realtime priority  0                      0
Max realtime timeout   unlimited               unlimited               us
```

可见, ss-server 最大可以打开32768个文件, 这个数字不算小吧, 那么ss-server现在打开了多少个文件呢

`/proc/{process_id}/fd` 是一个目录, 它为进程拥有的每个打开文件保存一个文件, 因此很容易计算我们达到限制的接近程度:

```
$ sudo ls /proc/6543/fd | wc -l
54
```

哇, ss-server 远没有达到打开文件数目的上限, 但是我们还是需要修改一下系统设置的, 因为系统中有多个用户, 有许多进程, 我们也要照顾到她们的需求是不是:

```
$ sudo vi /etc/security/limits.d/98-nofile.conf
# add lines to it
*      soft    nofile    512000
root   soft    nofile    512000
```

如果你的系统原来的硬限制值也较小, 那么可以这样设置:

```
$ sudo vi /etc/security/limits.d/98-nofile.conf
# add lines to it
*      soft    nofile    512000
*      hard    nofile    512000
root   soft    nofile    512000
root   hard    nofile    512000
```

开头的星号表示“将此规则应用于除root之外的所有用户”, 您可以猜测到root开头的规则 仅为root用户设置限制。最后的数字当然是您设置的新限制

网上的优化教程一般是直接修改 `/etc/security/limits.conf` 文件, 这样的做法并不好

重启系统, 检查是否生效:

```
$ ulimit -Sn
512000
$ ulimit -Sn
1048576
```

原来通过 shadowsocks-libev 上传文件比较慢, 原因可能在于BBR TCP加速需要打开更多的文件, 当打开文件数超过系统设置的上限时就会出错, 经设置98-nofile.conf后, 文件上传速度就很快了

RHEL, CentOS, Fedora, Scientific Linux用户还需要编辑 /etc/pam.d/login 文件并添加/修改以下行(确保获得pam_limits.so):




```
$ sudo vi /etc/pam.d/common-session

# add this line to it
session required pam_limits.so
```

相关资源:

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/ubuntu/etc/security/limits.d>
- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/ubuntu/etc/sysctl.d>
- <https://github.com/shadowsocks/shadowsocks/wiki/Optimizing-Shadowsocks>
- <https://www.cyberciti.biz/faq/linux-increase-the-maximum-number-of-open-files/>
- <https://software-download.name/2014/fanqiang-jiaocheng/>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07
Ubuntu server 最大打开文件数目优化

-  如何查看系统打开文件数量的限制
-  如何调整系统范围的文件描述符(FD)限制
-  如何设置用户级别打开文件描述符 FD 限制

Linux TCP UDP 网络性能优化

Shadowsocks 服务端系统网络性能优化的原则

- 在一开始, 能不优化的都不优化
- 可以在以后的使用中逐步测试、改进和增加优化选项

为什么对优化持谨慎的态度? 因为我们把多个优化选项放在同一个文件中, 如果带来了某方面的副作用, 给测试、排查带来了困难

最大队列大小优化

在通过TCP / UDP层处理数据之前, 系统会将数据放入内核队列中。net.core.netdev_max_backlog 值指定在传递到上层之前要放入队列的最大数据包数。对于高网络负载, 默认值是不够的, 因此简单地增加此值可以解决内核导致的性能损失问题。要查看默认值, 请将sysctl与\$ sysctl net.core.netdev_max_backlog一起使用。默认值为1000, 将其增加到3000以上将足以阻止数据包在10Gbps(或更多)网络中被丢弃

```
$ sysctl net.core.netdev_max_backlog
sysctl net.core.netdev_max_backlog = 1000

sudo vi /etc/sysctl.d/98-network-custom.conf
net.core.netdev_max_backlog = 4096
```

另一个类似的设置是 net.ipv4.tcp_max_syn_backlog 记住的连接请求的最大数量, 但仍未收到来自连接客户端的确认。对于内存超过128 MB的系统, 默认值为1024, 对于低内存计算机, 默认值为128。如果服务器过载, 请尝试增加此数量

```
$ sysctl net.ipv4.tcp_max_syn_backlog
net.ipv4.tcp_max_syn_backlog = 128

sudo vi /etc/sysctl.d/98-network-custom.conf
net.ipv4.tcp_max_syn_backlog = 4096
```

最大挂起连接数优化

应用程序可以在处理一个连接之前指定要放入队列的最大待处理请求数。当此值达到最大值时, 进一步的连接开始退出。对于发布大量连接的Web服务器等应用程序, 此值必须很高才能使这些连接正常工作

```
$ sysctl net.core.somaxconn
net.core.somaxconn = 128

$ sudo vi /etc/sysctl.d/98-network-custom.conf
net.core.somaxconn = 4096
```

TCP FIN超时优化

在TCP连接中, 双方必须独立关闭连接。Linux TCP发送FIN数据包以关闭连接并等待FINACK直到定义超时值

```
$ sysctl net.ipv4.tcp_fin_timeout
sysctl net.ipv4.tcp_fin_timeout = 60

$ sudo vi /etc/sysctl.d/98-network-custom.conf
net.ipv4.tcp_fin_timeout = 30
```

默认值(60)非常高, 可以减少到20或30以使TCP关闭连接并释放资源以进行另一个连接

重用 TIME_WAIT 状态的套接字进行新连接

根据[Linux文档](#), 您应该使用TCP_TW_REUSE 标志允许重新使用TIME_WAIT状态的套接字进行新连接

在处理必须处理TIME_WAIT状态下的许多短TCP连接的Web服务器时, 这似乎是一个不错的选择

```
$ sysctl net.ipv4.tcp_tw_reuse
```



```
net.ipv4.tcp_tw_reuse = 0

$ sudo vi /etc/sysctl.d/98-network-custom.conf
net.ipv4.tcp_tw_reuse = 1
```

tcp_keepalive_time 优化

TCP连接由两个套接字组成, 每个套接字在连接的两端。当一方想要终止连接时, 它会发送另一方确认的RST数据包并关闭其套接字

然而, 在此之前, 双方将无限期地保持其套接字开放。这使得一方可能有意或由于某些错误而关闭其插座, 而无需通过RST通知另一端。为了检测此场景并关闭过时连接, 使用TCP Keep Alive处理

有三个可配置属性可确定Keep-Alives的属性。在Linux上他们是1:

- tcp_keepalive_time
默认7200秒
- tcp_keepalive_probes
默认9
- tcp_keepalive_intvl
默认75秒

这个过程是这样的:

- 客户端打开TCP连接
- 如果tcp_keepalive_time秒的连接是静默的, 则发送一个空的ACK数据包
- 服务器是否使用自己的相应ACK进行响应?
 - 没有
 - 等待tcp_keepalive_intvl秒, 然后发送另一个ACK
 - 重复, 直到已发送的ACK探测数等于tcp_keepalive_probes
 - 如果此时未收到响应, 请发送RST并终止连接
 - 是: 返回第2步

默认情况下, 在大多数操作系统上启用了此处理过程, 因此一旦另一端无响应2小时11分钟(7200秒+ 75 * 9秒), 则会定期移除死TCP连接

```
$ sysctl net.ipv4.tcp_keepalive_time
net.ipv4.tcp_keepalive_time = 7200

$ sudo vi /etc/sysctl.d/98-network-custom.conf
net.ipv4.tcp_keepalive_time = 1200
```

启用智能MTU黑洞检测优化

一旦启用, 您的操作系统将尝试使用路径MTU发现机制在客户端和服务器之间找到MTU

您可以通过运行 `ip a` 检查接口上的MTU:

```
$ ip a | grep mtu
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq state UP group default qlen 1000
```

什么是 MTU

MTU = Maximum Transmission Unit

当Internet上的主机想要发送一些数据时, 它必须知道如何将数据分成数据包。特别是, 它需要知道数据包的最大大小。主机可以发送的数据包的最大大小称为 Maximum Transmission Unit, 最大传输单元, MTU

MTU越长, 性能越好, 但可靠性越差。这是因为丢失的数据包意味着要重新传输更多数据, 并且因为Internet上的许多路由器无法传输非常长的数据包

ICMP消息应该被传递给始发主机, 而主机应该调整该特定连接的MTU设置。此机制称为 Path MTU Discovery, 路径MTU发现

从理论上讲, 它很棒, 但遗憾的是, 在传送ICMP数据包时, 很多事情都可能出错。最常见的问题是由丢失ICMP数据包的防火墙配置错误引起的

路由器丢弃数据包但由于某种原因无法传递相关ICMP消息的情况称为 ICMP black hole, ICMP黑洞

当发生这种情况时，整个连接都会卡住。发送方不断尝试重新发送丢失的数据包，而接收方仅确认传送的小数据包

[RFC4821](#) 提出了一种检测ICMP黑洞的机制，并尝试以智能方式调整路径MTU。要在Linux类型上启用此功能，运行命令：

```
$ sysctl net.ipv4.tcp_mtu_probing
net.ipv4.tcp_mtu_probing = 0
$ sysctl net.ipv4.tcp_base_mss
net.ipv4.tcp_base_mss = 1024

$ sudo vi /etc/sysctl.d/98-network-custom.conf
net.ipv4.tcp_mtu_probing = 1
```

tcp_mtu_probing, 控制TCP分组化 - 层路径MTU发现。可选三个 值：

- 0 已禁用
- 1 默认情况下禁用，在检测到ICMP黑洞时启用
- 2 始终启用，使用tcp_base_mss的初始MSS

net.ipv4.tcp_base_mss 设置发现中使用的起始MSS值，如果系统默认小于1024，可改成1024：

```
$ sudo vi /etc/sysctl.d/98-network-custom.conf
net.ipv4.tcp_mtu_probing = 1
net.ipv4.tcp_base_mss = 1024
```

参考*：

- <https://blog.cloudflare.com/path-mtu-discovery-in-practice/>

可选优化：内核缓冲区优化

查看ubuntu 18.04系统默认的套接字缓冲区大小：

```
$ sudo sysctl net.core.wmem_default
net.core.wmem_default = 212992
$ sysctl net.core.rmem_default
net.core.rmem_default = 212992
$ sysctl net.core.rmem_max
sysctl net.core.rmem_max = 212992
$ sysctl net.core.wmem_max
net.core.wmem_max = 212992
```

212992 bytes, 换算成 KB, $212992 / 1024 = 208$ KB

这些参数显示分配给任何类型连接的默认和最大写入、读取缓冲区大小。由于分配的空间来自RAM，因此默认值设置总是有点低。增加这一点可能会提高运行NFS等服务器的系统的性能。将它们增加到256k / 4MB将最有效，否则您必须对这些值进行基准测试，以找到系统配置的理想值

我们把自定义网络优化都保存到 `/etc/sysctl.d/98-network-custom.conf`

```
$ sudo vi /etc/sysctl.d/98-network-custom.conf

# 256 KB / 4 MB
net.core.rmem_default = 262144
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_max = 4194304

# Or 256 Kb / 64 MB
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.core.rmem_max = 67108864
net.core.wmem_max = 67108864
```

67108864 bytes = 64 MB, 这个值是比较大的，最好是测试一下，对于你的系统，这个值是否是最优值

可选优化：TCP缓冲区大小优化

查看一下系统默认值：

```
$ sysctl net.ipv4.tcp_rmem
net.ipv4.tcp_rmem = 4096      87380  6291456
```

```
$ sysctl net.ipv4.tcp_wmem
net.ipv4.tcp_wmem = 4096      16384   4194304
```

这些值是三个整数的数组, 分别指定TCP读取和发送缓冲区的最小, 平均和最大值

注意: 值以页为单位。要查看页面大小, 请使用命令查看:

```
$ getconf PAGE_SIZE
4096
```

也就是设置的值必须是4096的倍数

TCP缓冲区最大值改成64 MB:

```
$ sudo vi /etc/sysctl.d/98-network-custom.conf
net.ipv4.tcp_rmem = 4096 87380 67108864
net.ipv4.tcp_wmem = 4096 16384 67108864
```

或者TCP缓冲区最大值改成12 MB:

```
net.ipv4.tcp_rmem = 4096 87380 12582912
net.ipv4.tcp_wmem = 4096 16384 12582912
```

有的人推荐TCP缓冲区最大值为 4MB:

```
net.ipv4.tcp_rmem = 4096 87380 4194304
net.ipv4.tcp_wmem = 4096 16384 4194304
```

Linux 2.6 内核开始, 有一个自动调整功能, 可以动态调整TCP缓冲区大小, 直到达到最大值。默认情况下, 此功能处于启用状态, 我建议将其保持打开状态。您可以通过运行以下命令来检查它:

```
$ cat /proc/sys/net/ipv4/tcp_moderate_rcvbuf
1
```

要在它关闭的情况下临时打开它, 请使用下面给出的命令:

```
sysctl -w net.ipv4.tcp_moderate_rcvbuf = 1
```

如果您发现内核缓冲区是您的瓶颈, 需要增加最大缓冲区大小, 则此设置将空间分配为最大值。无需更改平均值, 但必须将最大值设置为高于BDP(带宽延迟乘积)以获得最大吞吐量

$BDP = \text{带宽 (B/秒)} * RTT(\text{秒})$, 其中RTT(往返时间)可以通过ping到任何其他系统来计算, 并以秒为单位查找平均时间

此项如上优化后可能造成上传文件失败或变慢, 故列为可选优化

可选优化: Time Wait优化

TIME WAIT TCP套接字状态是套接字关闭但等待处理仍在网络中的数据包的狀態。参数tcp_max_tw_buckets是 TIME_WAIT 状态下的最大套接字数。达到此数字后, 系统将开始在此状态下销毁套接字

此限制仅用于防止简单的DoS攻击, 您不得人为地降低限制, 而是增加它(可能在增加安装的内存之后), 如果网络条件需要超过默认值

```
$ sysctl net.ipv4.tcp_max_tw_buckets
net.ipv4.tcp_max_tw_buckets = 4096

$ sudo vi /etc/sysctl.d/98-network-custom.conf
net.ipv4.tcp_max_tw_buckets = 5000
```

如果遇到大量的TCP 错误, 如:

```
__ratelimit: 33491 callbacks suppressed
TCP: time wait bucket table overflow
```

可以增加 `net.ipv4.tcp_max_tw_buckets` 的值, 比如 654320, 前提是拥有足够的内存

请尝试以下命令来确定您是否有来自一个地址的大量连接, 或者您是否受到分布式攻击

```
netstat -nt | cut -c 40- | cut -d: -f1 | sort | uniq -c | sort -n netstat -nt | cut -d: -f2 | sort | uniq -c | sort -n
```

如果您从几个IP地址获得高数字, 则更容易限制连接。然后, 您可以向 iptables 添加拒绝规则或速率限制规则, 以限制从这些地址访问经测试, 优化此项可能造成上传文件至某些网站超时或错误

可选优化: IP端口范围优化

net.ipv4.ip_local_port_range 显示可用于新连接的所有端口。如果没有空闲端口, 则连接将被取消。增加此值有助于防止此问题

如果您的Linux服务器正在打开大量传出网络连接, 则需要增加本地端口范围。默认情况下范围很小。例如, 如果squid代理服务器用完了端口, 它就会受到攻击。其他示例包括繁忙的流量网络服务器, 如nginx负载均衡器, LXD vm等

我们可以加大用于新连接的端口选择范围, 但是有一个风险, 某个程序使用的特定端口可能被该服务器上的其他程序随机选取源端口给占用掉了, 解决办法是将服务监听的特定端口以逗号分隔全部添加到ip_local_reserved_ports中

ip_local_reserved_ports 逗号分隔范围列表指定为已知第三方保留的端口应用。自动端口不会使用这些端口 赋值(例如, 当使用port调用 connect()或bind()时 数字0)。显式端口分配行为保持不变

用于输入和输出的格式是逗号分隔范围列表, 例如保留端口1,2,3,4和1可以这样指定: 1,2-4,10-10

写入文件后将清除以前保留的所有内容端口并使用中给出的端口更新当前列表输入

请注意 ip_local_port_range 和 ip_local_reserved_ports 设置是独立的, 并且都由内核考虑确定哪些端口可用于自动端口时分配

本翻墙方案 <https://github.com/softwaredownload/openwrt-fanqiang> shadowsocks-libev 服务端指定监听 1098 端口, 1098 应该加入到 ip_local_reserved_ports 这样 ip_local_port_range 无论怎么设置都不影响 shadowsocks-libev 监听在 1098 端口

```
$ sysctl net.ipv4.ip_local_port_range
net.ipv4.ip_local_port_range = 32768 60999
$ sysctl net.ipv4.ip_local_reserved_ports
# 默认空值

$ sudo vi /etc/sysctl.d/98-network-custom.conf
net.ipv4.ip_local_port_range = 10000 65535
net.ipv4.ip_local_reserved_ports = 1098
```

以上定义了TCP和UDP使用的本地端口范围选择本地端口。第一个数字是第一个, 第二个是最后一个本地端口号。如果可能, 这些数字具有不同的奇偶校验即一个偶数和一个奇数值是更好的。默认值分别为32768和60999, 或者由发行版或系统管理员设置的默认值。在此示例中, 1024 不是奇数, 65535是奇数

- <https://www.cyberciti.biz/tips/linux-increase-outgoing-network-sockets-range.html>

无需优化

- net.ipv4.tcp_syncookies

Ubuntu 18.04 以下已经默认设置:

```
net.ipv4.tcp_syncookies = 1
```

- net.ipv4.tcp_tw_recycle

在 Linux 内核 4.12 开始已经移除这个选项了, ubuntu 18.04 不需要设置此值。如果你的内核较旧, 可以增加设置:

```
net.ipv4.tcp_tw_recycle = 0
```

/etc/sysctl.d/98-network-custom.conf

```
net.core.netdev_max_backlog = 4096
net.ipv4.tcp_max_syn_backlog = 4096
net.core.somaxconn = 4096

net.ipv4.tcp_fin_timeout = 30

net.ipv4.tcp_tw_reuse = 1

net.ipv4.tcp_keepalive_time = 1200
```

```
net.ipv4.tcp_mtu_probing = 1
```

使 Linux 系统网络优化立即生效：

```
sudo sysctl --system
```

需要注意的是，上面的TCP/UDP优化只是供参考，最佳的设置需要各人自己测试才能确定，设置不当可能会有问题，比如说使上传视频到外网变慢

在Ubuntu 18.04 系统上，我们经过逐项对照系统的默认值，得到上面的优化设置。如果你的系统不是 Ubuntu 18.04，可以在 [Digital Ocean](#) 创建一个新的 VPS，默认就是最新的 Ubuntu 系统，于就是可以按照本 [科学上网教程](#) 的方案进行系统优化

相关资源：

- <https://github.com/softwaredownload/openwrt-fanqiang/tree/master/ubuntu/etc/sysctl.d>
- <https://github.com/shadowsocks/shadowsocks/wiki/Optimizing-Shadowsocks>
- <https://opensourceforu.com/2016/10/network-performance-monitoring/>
- <https://software-download.name/2014/fanqiang-jiaocheng/>
- <https://fanqiang.software-download.name/>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-12-07

[Linux TCP UDP 网络性能优化](#)

- 🏔 Shadowsocks 服务端系统网络性能优化的原则
- 🏠 最大队列大小优化
- 🏠 最大挂起连接数优化
- 🏠 TCP FIN超时优化
- 🏠 重用 TIME_WAIT 状态的套接字进行新连接
- 🏠 tcp_keepalive_time 优化
- 🏠 启用智能MTU黑洞检测优化
- 🏠 可选优化：内核缓冲区优化
- 🏠 可选优化：TCP缓冲区大小优化
- 🏠 可选优化：Time Wait优化
- 🏠 可选优化：IP端口范围优化
- 🏠 无需优化
- 😊 /etc/sysctl.d/98-network-custom.conf

Linux Ubuntu swap 交换文件优化

我们在 [Digital Ocean](#) 创建 VPS (Droplet) 时, 最便宜的配置如下:

Choose a size

Standard Droplets

Balanced virtual machines with a healthy amount of memory tuned to host and scale applications like blogs, web applications, testing / staging environments, in-memory caching and databases.

MEMORY	vCPUs	SSD DISK	TRANSFER	PRICE
1 GB	1 vCPU	25 GB	1 TB	\$5/mo \$0.007/hr
2 GB	1 vCPU	50 GB	2 TB	\$10/mo \$0.015/hr

内存是 1GB, 一般情况下是够用了, 如果多开几个 shadowsocks 进程, 多个用户同时上 youtube.com, 那么内存可能会不够用, 怎么办
可以设置 swap 交换文件。DO 硬盘都是 SSD, 内存不够时可以使用硬盘的swap, 速度也不错

注意:

尽管通常建议对使用传统旋转硬盘驱动器的系统进行交换, 但使用SSD交换可能会导致硬件随着时间的推移而出现问题。出于这种考虑, 通常不建议在使用SSD存储的提供商上启用交换文件。这样做会影响您和您的邻居的底层硬件的可靠性

检查系统是否设置过 swap

在开始之前, 我们可以检查系统是否已经有一些可用的交换空间。可以有多个交换文件或交换分区, 但通常一个就足够了

我们可以通过键入以下内容来查看系统是否已配置了交换文件

```
sudo swapon - show
```

如果您没有收到任何输出, 这意味着您的系统当前没有可用的交换空间

您可以使用 free 验证没有活动交换空间:

```
free -h
```

检查硬盘驱动器分区上的可用空间

为交换分配空间的最常用方法是使用专用于该任务的单独分区。但是, 改变分区方案并不总是可行的。我们可以轻松创建驻留在现有分区上的交换文件

在我们这样做之前, 我们应该键入以下内容来检查当前磁盘使用情况:

```
df -h
```

有足够的可用空间的时候, 我们才能创建 swap 文件

创建 swap 交换文件

现在我们知道有了可用的硬盘空间, 我们可以在文件系统中创建一个交换文件。我们将在根 (/) 目录中创建一个我们想要的交换大小的文件

创建交换文件的最佳方法是使用fallocate程序。此命令立即创建预分配大小的文件

由于我们示例中的服务器具有1024MB的RAM, 因此我们将在本教程中创建3 GB的文件。调整此项以满足您自己的服务器的需求:

```
sudo fallocate -l 3G /swap
```

我们可以通过输入以下内容来验证是否保留了正确的空间量:

```
$ ls -lh /swap
-rw-r--r-- 1 root root 3.0G Dec 19 11:14 /swap
```

启用 swap 交换文件

现在我们有一个正确大小的文件, 我们需要实际将其转换为交换空间

首先, 我们需要锁定文件的权限, 以便只有具有root权限的用户才能读取内容。这可以防止普通用户访问该文件, 这会产生重大的安全隐患

通过键入以下内容使该文件只能由root访问:

```
sudo chmod 600 /swap
```

键入以下命令验证权限更改:

```
$ ls -lh /swap
-rw ----- 1 root root 3.0G Dec 19 11:14 /swap
```

如您所见, 只有root用户启用了读写标志

我们现在可以通过键入以下内容将文件标记为交换空间

```
sudo mkswap /swap
```

标记文件后, 我们可以启用交换文件, 允许我们的系统开始使用它:

```
sudo swapon /swap
```

我们可以通过输入以下内容来验证交换是否可用:

```
sudo swapon - show
```

可以再次检查:

```
$ free -h
              total        used        free      shared  buff/cache   available
Mem:           985M          96M         113M          1.6M          775M          727M
Swap:           3.0G          268K          3.0G
```

使 swap 交换文件永久化

我们最近的更改已启用当前会话的交换文件。但是，如果我们重新启动，服务器将不会自动保留交换设置。我们可以通过将交换文件添加到 `/etc/fstab` 文件来更改此设置

备份 `/etc/fstab` 文件，以防出现任何问题：

```
sudo cp /etc/fstab /etc/fstab.bak
```

您可以通过键入以下内容将交换文件信息添加到 `/etc/fstab` 文件的末尾：

```
echo 'swap none swap sw 0 0' | sudo tee -a /etc/fstab
```

调整swap交换文件设置

您可以配置一些选项，这些选项会在处理交换时对系统的性能产生影响

调整 Swappiness 属性

swappiness参数配置系统将数据从RAM交换到交换空间的频率。这是介于0和100之间的值，表示百分比

值接近于零时，除非绝对必要，否则内核不会将数据交换到磁盘。请记住，与交换文件的交互是“昂贵的”，因为它们比与RAM的交互花费更长的时间，并且它们可能导致性能的显著降低。告诉系统不要太依赖交换通常会使您的系统更快

接近100的值将尝试将更多数据放入交换中以努力保持更多RAM空间。根据应用程序的内存配置文件或服务器的使用情况，在某些情况下可能会更好

我们可以通过输入以下内容来查看当前的swappiness值：

```
$ cat /proc/sys/vm/swappiness
60
```

对于桌面，swappiness设置为60并不是一个糟糕的值。对于服务器，您可能希望将其移近0

我们可以使用sysctl命令将swappiness设置为不同的值

例如，要将swappiness设置为20，我们可以输入：

```
$ sudo sysctl vm.swappiness = 20
vm.swappiness = 20
```

要使设置在重启系统后仍然生效，建议把自定义设置保存到 `/etc/sysctl.d` 目录下

```
$ sudo vi /etc/sysctl.d/98-swap.conf
vm.swappiness = 20
```

调整 Cache Pressure 缓存压力设置

您可能想要修改的另一个相关值是 `vfs_cache_pressure`。此设置配置系统将选择多少缓存inode和dentry信息而不是其他数据

基本上，这是关于文件系统的访问数据。这通常是非常昂贵的查询和非常频繁的请求，所以这是你的系统缓存的一个很好的事情。您可以通过再次查询proc文件系统来查看当前值：

```
$ cat /proc/sys/vm/vfs_cache_pressure
100
```

由于它当前已配置，我们的系统会过快地从缓存中删除inode信息。我们可以通过键入以下内容将其设置为更保守的设置（如50）：

```
$ sudo sysctl vm.vfs_cache_pressure = 50
vm.vfs_cache_pressure = 50
```

同样，我们要把这个设置保存到 `/etc/sysctl.d/98-swap.conf` 文件中：

```
$ sudo vi /etc/sysctl.d/98-swap.conf
vm.vfs_cache_pressure = 50
```


我们把自定义内核参数设置都保存到 `/etc/sysctl.d/` 目录下, 并且文件名以 98 开头, 当我们创建一个新的 VPS 时, 可以轻松地用 `tar` 命令把 `/etc/sysctl.d/98*` 文件打包并迁移到新的环境中。你可以在下面地址查看内核参数优化文件:





<https://github.com/softwaredownload/openwrt-fanqiang/tree/master/ubuntu/etc/sysctl.d>

相关资源:

- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/ubuntu/etc/sysctl.d/98-swap.conf>
- <https://software-download.name/2014/fanqiang-jiaocheng/>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

[Linux Ubuntu swap 交换文件优化](#)

-  检查系统是否设置过 swap
-  检查硬盘驱动器分区上的可用空间
-  创建 swap 交换文件
-  启用 swap 交换文件
-  使 swap 交换文件永久化
-  调整 swap 交换文件设置
-  调整 Cache Pressure 缓存压力设置

附录

翻墙常用资源及如何贡献本项目

最简单的路由器刷OpenWrt翻墙方案:

- <https://github.com/softwaredownload/openwrt-fanqiang>

在线阅读OpenWrt路由器翻墙、科学上网教程:

- <https://fanqiang.software-download.name>
- <https://github.com/softwaredownload/openwrt-fanqiang/blob/master/SUMMARY.md>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-10-22

翻墙教程资源汇总

翻墙软件

- [Shadowsocks.org](#)
- [Shadowsocks libev](#)
- [trojan](#)
- [Simple Obfs](#)
- [OpenWrt Simple Obfs](#)
- [Shadowsocks Windows](#)
- [Shadowsocks QT5](#)
- [Shadowsocks Android](#)
- [Shadowsocks GO](#)
- [V2Ray 模块化的代理软件包](#)
- [Obfuscated OpenSSH Patch by zinglau](#)
- [XX-Net 接力GoAgent](#)
- [Outline](#) (Google 开发, 因Google有收集隐私的嗜好, 一般不推荐使用)

翻墙方案

- [OpenWrt路由器智能自动透明翻墙、科学上网教程](#)

DNS 相关

- [Dnscrypt proxy](#)
- [Pcap DNSProxy](#)
- [ChinaDNS](#)
- [dnsmasq China List](#)
- [dnsforwarder](#)
- [A DNS server/forwarder/dispatcher written in Go](#)
- [glider - forward proxy with multiple protocols support](#)
- <https://www.rubyfish.cn/server>
 - 115.159.154.226
 - 47.99.165.31
- <https://www.hixns.com>
 - 40.73.101.101
- <https://dns.xsico.cn>
 - 182.254.242.15
- 中科大 DNS, 端口53, 5353, 支持TCP查询
 - 202.38.93.153 (教育网)
 - 202.141.176.93 (中国移动)
 - 202.141.162.123 (中国电信)
- [Free Public DNS](#)

OpenWrt教程

- [跟hoowa学做智能路由](#)
- [跟 UMU 一起玩 OpenWRT](#)

其他软件

- <https://git-scm.com/download/win>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name>

2019-07-03

翻墙教程资源汇总

-  翻墙软件
-  翻墙方案
-  DNS 相关
-  OpenWrt教程
-  其他软件

本地阅读本教程的方法

git clone项目

```
cd ~/Downloads
git clone https://github.com/softwaredownload/openwrt-fanqiang
```

下载Markdwon阅读软件 Typora

Typora有个神奇的地方, 就是 Markdown 写作和预览是一体的这就避免了多数 Markdown 写作软件会有的尴尬:边写作边预览时, 屏幕宽度始终不够

下载后, 点击菜单 File 选择 Open Folder... 选择 fanqiang/ebook

点击左边的导航栏切换内容



你是个有爱心的人, 阅读了本教程, 想要回馈这个开源项目, 在阅读时顺便修改一些错字, 加进补充内容, 增加一章你的路由器应用本教程翻墙的过程等等, 然后提交 pull request

相关资源:

- <https://software-download.name/2014/fanqiang-jiaocheng/>
- <https://typora.io>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

本地阅读本教程的方法

-  git clone项目
-  下载Markdwon阅读软件 Typora

知识若不分享，实在没有意义

▲ 这个世界为什么圣人这么少？

人类历史上存在过无数人，他们都不见了，他们都到哪里去了，他们曾有过什么样的故事，可曾有人在想起他们的笑容？通过历史书，我们知道了历史上存在过的一些人物的名字，其中少数人，为人类的发展作出了特别的贡献，我们可以称他们为圣人，这样的人，一只手就数得过来

历史上存在过的人这么多，为什么圣人却这么少？

我认为，这是因为，普通人的一生，主要是在思考怎么得到更多，而较少想到去付出。得到越多越好，付出越少越好，这就是普通人

圣人是怎么样的，是不是只想着付出，不计收获？不是的，我认为圣人是付出得到比较均衡的人。只付出而不得到，自己就很快会陷入困境，就没有能力去帮助更多人

圣人得到什么，就会想着怎么样去回馈外界，回馈社会，在回馈过程中自己得到快速成长，从而有更大的能力去回馈更多，圣人于是逐渐长成

我这么说，并不是希望谁成为圣人。圣人并不知道自己是圣人，也不会去想这个事情。有一个信念，就要去实行，生命的意义就在于点滴的行动，能做多少就做多少，当生命之花最终凋落时，我们得到的都将失去，我们付出的也许还会存在于这个世界很长的时间

🧑 我为什么写这个教程

生在天朝，上网各种不方便，很是苦恼，什么OpenWrt，没有听说过，不知道哇。上网查相关论坛，非注册用户附件下载隐藏，图片隐藏，各种限制。也有一些教程散布在网上，需要自己整合。终于，花了N个白天，给家里的路由器翻墙了。我是个习惯于换位思考的人，想想自己花了很多时间查各种资料，何不花时间整合各种资源并加上自己的心得，写成系列教程，公布在网上？

于是，又是N个白天(N > 10)，学习Git, GitHub, GitBook, Ubuntu, Markdown, OpenWrt, 各种调试、编译。经常一天的绝大部分时间在写这个教程。钱可以少赚些，当下够用就行，这个教程还得认真写，没有想过要得到什么，只是觉得白发已生，人生不能虚度，给这个世界留下一些自己的印记也总是好的。虽然不对别人说，但也未尝不可在人少时偷偷笑一声，并对自己说：我这样的好人，在这个世界上可是不多呢，哈哈

🧑 为什么以开源方式发布在GitHub

为什么不写在博客上呢？如果写在博客上，就要自己维护博客，一直维护下去总是个麻烦事。GitHub总比自己维护的博客稳定，或者说能存在更长时间。即使GitHub倒闭，也就一个git命令就可以托管到其他网站，何况GitHub至少现在看来是来日方长呢

开源方式发布，更是希望阅读本教程翻墙成功的朋友，如果你的路由器型号不被本教程覆盖，就写下自己的翻墙实践过程，提交到本项目中，以帮助相关朋友。我在教程中以 D-Link DIR-505为范例，演示了如何参与到本项目中来，将在下一节详述

相关资源：

- <https://software-download.name/2014/fanqiang-jiaocheng/>
- <https://fanqiang.software-download.name/>

版权所有，转载请注明出处：<https://fanqiang.software-download.name> 2018-12-07

知识若不分享，实在没有意义

- ▲ 这个世界为什么圣人这么少？
- 🧑 我为什么写这个教程
- 🧑 为什么以开源方式发布在GitHub

如何贡献本项目

虽然说原理是通用的, 本教程内容可以应用到绝大多数路由器中去。然而, 高手毕竟少数, 多数有翻墙需求的人可能都没有用过Linux系统, 没有听说过OpenWrt, 针对他们, 最好是一种路由器类型(型号)一个教程。并且最好提供预编译的固件, 刷上这个预编译的固件后, 修改极少的参数就可以自动翻墙

在你应用本教程原理翻墙的过程中, 把详细应用过程一步步写下来, 并贡献到本项目中, 以帮助更多的人

假如你的路由器是 netgear wndr3800



如何通过 Github 贡献本项目:

先阅读 Github [贡献向导](#), 然后:

- Fork 本项目 (<https://github.com/software-download/openwrt-fanqiang/fork>)
- 创建你的分支 (git checkout -b my-new-feature)
- 提交你的改进 (git commit -am 'Add some feature')
- Push到你的分支 (git push origin my-new-feature)
- 到github.com 创建 Pull Request



如何为新的路由器创建翻墙教程:

```
cd openwrt-fanqiang
mkdir -p ebook/wndr3800/images
mkdir openwrt/wndr3800
```

在ebook目录下创建以路由器型号为名的目录, 以wndr3800为例, 教程在ebook/wndr3800目录下, 图片在wndr3800/images在目录下

wndr3800专用的配置文件在openwrt/wndr3800下, 注意, openwrt/default目录已有的配置文件可以省略

路径、文件名都小写, 因Windows系统是大小写不敏感的

在你的教程中最好提供预编译固件的稳定下载地址。如果你没有稳定的下载空间, 可以提交一个issue, 附上临时下载地址, 我会上传到稳定下载地址, 然后你可以修改教程加上稳定下载地址。注意教程目录下不要直接包含固件文件, 大的二进制文件不需要用git跟踪

你可以用LiteIDE写教程

修改目录文件, openwrt/SUMMARY.md, 把你的教程作为新的一章, 放在最后一章之前

如果你的路由器型号与教程中的相同或类似, 你也可以参与到本项目中来, 你可以修正错误, 补充不详细的地方, 文字润色, 提出建议等

相关资源:

- <https://software-download.name/2014/fanqiang-jiaocheng/>
- <https://fanqiang.software-download.name/>

版权所有, 转载请注明出处: <https://fanqiang.software-download.name> 2018-12-07

[如何贡献本项目](#)

- [如何通过 Github 贡献本项目:](#)
- [如何为新的路由器创建翻墙教程:](#)