**centos7下shadowsocks服务端的安装配置与自启动（VPN)**

[原文](https://www.lexsion.com/index.php/archives/168/)

20221126 0:12测试通过 centos7.9

前言：

# CentOS7下编译安装Shadowsocks-libev

## 前言：

Shadowsocks是一种基于[Socks5](https://zh.wikipedia.org/wiki/SOCKS" \l "SOCK5)代理方式的加密传输协议，也可以指实现这个协议的各种开发包。在中国大陆，本工具广泛用于解决长城防火墙的未知错误而导致的Google、YouTube、Facebook等正常境外网站不能正常访问。网上有大量热心网友提供了各种版本的一键安装脚本，用于支持那些非计算机从业人员快速方便的在服务器上搭建服务端。但是某些情况下，一键脚本的安全性难以保证；另一方面，使用一键脚本不能让我们具体了解到这个工具是如何工作的。而本文则提供了一步一步从源代码手动编译安装Shadowsocks-libev的过程。

**阅读本文，您可能需要基本的Linux系统操作技术与网络技术基础。**

## 步骤：

1. 准备编译环境，在CentOS7下运行以下两条命令完成环境准备。
2. yum install epel-release -y

yum install gcc gettext autoconf libtool automake make pcre-devel asciidoc xmlto c-ares-devel libev-devel libsodium-devel mbedtls-devel -y

Shell

Copy

1. 安装Git，并从GitHub下载源码。
2. yum install git -y
3. cd /usr/local/src

git clone https://github.com/shadowsocks/shadowsocks-libev.git

Shell

Copy

1. 配置并编译源代码。
2. cd shadowsocks-libev
3. git submodule update --init --recursive
4. sh autogen.sh
5. ./configure --disable-documentation
6. make

make install

1. 配置Shadowsocks服务端配置文件，文件位于 .../shadowsocks-libev/debian/config.json,将此文件复制到/etc/shadowsocks-libev/config.json
2. mkdir /etc/shadowsocks-libev

cp ./debian/config.json /etc/shadowsocks-libev/config.json

1. 使用vi编辑器打开/etc/shadowsocks-libev/config.json。

vi /etc/shadowsocks-libev/config.json

1. 按照自己的需求填写服务端参数。
2. {
3. "server":"0.0.0.0", //服务端IP地址,Linux系统下0.0.0.0表示本地
4. "server\_port":8388, //服务端端口号
5. "local\_port":1080, //本地端口号
6. "password":"barfoo!", //密码
7. "timeout":10, 超时时长（秒）
8. "method":"chacha20-ietf-poly1305" //加密方式
9. }
10. #加密方式据说有如下几种：
11. aes-256-gcm
12. aes-192-gcm
13. aes-128-gcm
14. aes-256-ctr
15. aes-192-ctr
16. aes-128-ctr
17. aes-256-cfb
18. aes-192-cfb
19. aes-128-cfb
20. camellia-128-cfb
21. camellia-192-cfb
22. camellia-256-cfb
23. chacha20-ietf-poly1305
24. chacha20-ietf
25. chacha20

rc4-md5

1. 将shadowsocks-libev.default文件复制到/etc/sysconfig/shadowsocks-libev。  
   将shadowsocks-libev.service文件复制到/usr/lib/systemd/system/目录。因为本例中编译安装的程序文件位置为/usr/local/bin/，需要修改一下文件中的启动命令。
2. cp /usr/local/src/shadowsocks-libev/rpm/SOURCES/systemd/shadowsocks-libev.default /etc/sysconfig/shadowsocks-libev
3. cp /usr/local/src/shadowsocks-libev/rpm/SOURCES/systemd/shadowsocks-libev.service /usr/lib/systemd/system/

vi /usr/lib/systemd/system/shadowsocks-libev.service

将以下一行：

ExecStart=/usr/bin/ss-server -c "$CONFFILE" $DAEMON\_ARGS

替换成以下内容，按ESC键输入:wq，按Enter键保存退出即可。

ExecStart=/usr/local/bin/ss-server -c "$CONFFILE" $DAEMON\_ARGS

1. 运行服务，检查服务是否正常运行，若服务运行正常，即显示：Active: active (running)。
2. systemctl start shadowsocks-libev

systemctl status shadowsocks-libev

1. 确定服务正常后，将服务添加到开机启动。这时我们在客户端中输入服务器的IP地址等配置信息后就可以使用了，若依然不可以，请检查配置信息是否填写正确，并检查服务端防火墙配置。

systemctl enable shadowsocks-libev

## 防火墙配置：

若服务端与客户端配置均正确，依然无法建立连接，我们便需要检查是否服务端防火墙导致。若服务器系统配置了防火墙并已经启动，那么在配置防火墙开放对应服务端口前，自然是无法与服务器建立连接的。其实我们作为服务器的管理人员，自然应该知道自己服务器是否有开启防火墙的。

1. 我们可以使用[此网站](https://www.yougetsignal.com/tools/open-ports/)检查某个端口是否可以访问。输入IP地址和端口号后点击Check即可检测。若提示Port XX is open on x.x.x.x 即表示端口已经打开。反之 close即表示端口没有反应。此时我们进一步检查防火墙。
2. 检查防火墙服务是否运行，CentOS7默认将原来CentOS6时代的防火墙iptables升级为了firewalld。使用以下命令检查防火墙运行状态。若回显信息提示：running，则表示防火墙运行中。

firewall-cmd --state

1. 为Shadowsocks在防火墙中添加规则，开放对应端口的TCP和UDP通信。下以上文配置文件中的8388端口为例，具体按自己需求修改。到此为止，我们便完成了防火墙的配置。
2. firewall-cmd --zone=public --add-port={8388/tcp,8388/udp} --permanent

firewall-cmd --reload

本文到此结束。下附Ubuntu安装：

# Ubuntu下编译安装Shadowsocks-libev

### 安装各种编译需要的库、工具等：

sudo apt install --no-install-recommends build-essential autoconf libtool libssl-dev gawk debhelper dh-systemd init-system-helpers pkg-config asciidoc xmlto apg libpcre3-dev zlib1g-dev libev-dev libudns-dev libsodium-dev libmbedtls-dev libc-ares-dev automake

Shell

Copy

### 拉取源代码：

sudo apt install git

git clone https://github.com/shadowsocks/shadowsocks-libev.git

cd shadowsocks-libev

git submodule update --init

./autogen.sh && ./configure --disable-documentation && make

sudo make install

Shell

Copy

### 创建配置文件：

创建配置文件目录：

sudo mkdir /etc/shadowsocks-libev

Shell

Copy

**单用户配置文件：**

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

Shell

Copy

{

"server":"0.0.0.0",

"server\_port":8388,

"local\_port":1080,

"password":"mypassword",

"timeout":300,

"method":"aes-256-gcm" ,

"fast\_open": false

}

JSON

Copy

**多用户配置文件：**

sudo vi /etc/shadowsocks-libev/manager\_config.json

Shell

Copy

{

"server":"0.0.0.0",

"local\_port":1080,

"method":"aes-256-gcm" ,

"timeout":300,

"fast\_open": true,

"port\_password": {

"8388": "barfoo1",

"8389": "barfoo2"

}

}

JSON

Copy

### 创建Systemd服务文件：

**单用户：**

sudo vi /etc/systemd/system/shadowsocks-libev.service

Shell

Copy

[Unit]

Description=Shadowsocks-libev Server

After=network.target

[Service]

Type=simple

ExecStart=/usr/local/bin/ss-server -c /etc/shadowsocks-libev/config.json -u

Restart=on-abort

[Install]

WantedBy=multi-user.target

Config

Copy

**多用户：**

[Unit]

Description=Shadowsocks-libev Server

After=network.target

[Service]

Type=simple

ExecStart=/usr/local/bin/ss-manager -c /etc/shadowsocks-libev/manager\_config.json -u

Restart=on-abort

[Install]

WantedBy=multi-user.target

Config

Copy

### 启动并设置开机自启：

sudo systemctl start shadowsocks-libev

sudo systemctl enable shadowsocks-libev

Shell

Copy

## 参考：

1. [shadowsocks-libev-GitHub项目页](https://github.com/shadowsocks/shadowsocks-libev)

标签: none

**ssr客户端下载**

Windows SSR客户端 [点击下载地址](https://github.com/shadowsocksr-backup/shadowsocksr-csharp/releases)  
MacOS SSR客户端 [点击下载地址](https://github.com/shadowsocksr-backup/ShadowsocksX-NG/releases)  
Linux SSR客户端 [点击下载地址](https://github.com/erguotou520/electron-ssr/releases)  
安卓 SSR客户端 [点击下载地址](https://github.com/shadowsocksr-backup/shadowsocksr-android/releases/download/3.4.0.8/shadowsocksr-release.apk)

苹果手机SSR客户端：Potatso Lite、Potatso、shadowrocket都可以作为SSR客户端，但这些软件目前已经在国内的app商店下架，可以用美区的appid账号来下载。但是，如果你配置的SSR账号兼容SS客户端，或者协议选择origin且混淆选择plain，那么你可以选择苹果SS客户端软件(即协议和混淆可以不填)，APP商店里面有很多，比如：openwingy、superwingy、bestwingy、wingy 、greatwingy等。