————————————————————脚本1————————————————————

```
#!/bin/sh

# Yum update
yum update -y                    #更新系统

# Install python setup tools
yum install python-setuptools -y        #安装pip

# Install pip
easy_install pip

# Install shadowsocks
pip install shadowsocks
pip install -U shadowsocks

# Create shadowsocks.json
cat <<EOF > /etc/shadowsocks.json               #将本应重定向到标准输出的内容重定向到/
etc/shadowsocks文件中
{
    "server": "0.0.0.0",
    "server_port": 8888,
    "local_port": 1080,
    "local_address": "127.0.0.1",
    "password": "www.qttc.net",
    "method": "aes-256-cfb",
    "timeout": 600
}
EOF

# Start firewall
systemctl enable firewalld              #开启firewall防火墙
systemctl start firewalld

# Add port for firewall
firewall-cmd --permanent --add-port=8888/tcp    #添加端口到防火墙
firewall-cmd --permanent --add-port=8888/udp
firewall-cmd --reload

# Install supervisor
easy_install supervisor                                 #安装supervisor，这是个后台进程
管理用的
echo_supervisord_conf > /etc/supervisord.conf

# Write configure to supervisord.conf
cat <<EOF >> /etc/supervisord.conf              #配置supervisor
[program:ss]
command=/usr/bin/ssserver -c /etc/shadowsocks.json
redirect_stderr=true
user=root
stdout_logfile=/tmp/ss.log
EOF

# Start supervisord
/usr/bin/supervisord -c /etc/supervisord.conf           #启动

# Setup boot start
echo "/usr/bin/supervisord -c /etc/supervisord.conf" >> /etc/rc.local   #将启动命令写
入rc.local

# Create check.sh
cat <<EOF > /root/check.sh                      #检查是否启动成功
#!/bin/sh
COUNT=\`/bin/ps -ef | grep supervisord | grep -v 'grep' | wc -l\`       #将命令的返回
值赋值给COUNT,这个返回值应该就是执行结果
```

```
if [ \$COUNT == 0 ];
then                                                                    #执行上述命令，如果返回值为0，则表示启动失败，重新启动
    echo "Starting..."
    /usr/bin/kill -9 \`/usr/bin/ps -ef | grep ssserver | grep -v 'grep' | /usr/bin/awk  '{print $2}'\`             #关闭相关程序
    /bin/rm -rf /tmp/supervisor.sock                                    #删除supervisor遗留内容
    /usr/bin/supervisord -c /etc/supervisord.conf                       #再启动一次
fi
EOF

chmod +x /root/check.sh

# Add check.sh to crontab
crontab -l > /root/mycron                #备份目前的crontab时间表
echo "* * * * * /bin/sh /root/check.sh" >> /root/mycron          #追加check.sh文件到时间表中
crontab /root/mycron                                #提交/root/mycron文件给crontab进程
rm -rf /root/mycron                                                     #删除/root/mycron文件

# Optimize system
echo '* soft nofile 51200' > /etc/security/limits.conf          #这个应该是防火墙的设置
echo '* hard nofile 51200' > /etc/security/limits.conf

ulimit -n 51200                                     #设置打开的文件描述符数量

/sbin/modprobe tcp_hybla                            #载入tcp_hybla模块，可以加入美国VPS的访问

cat <<EOF > /etc/sysconfig/modules/hybla.modules        #将加载命令写入文件中
#!/bin/sh
/sbin/modprobe tcp_hybla
EOF

chmod +x /etc/sysconfig/modules/hybla.modules           #给予可执行权限

cat <<EOF >> /etc/sysctl.conf                       #一些设置性的内容
fs.file-max = 51200
net.core.rmem_max = 67108864
net.core.wmem_max = 67108864
net.core.netdev_max_backlog = 250000
net.core.somaxconn = 4096
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_tw_recycle = 0
net.ipv4.tcp_fin_timeout = 30
net.ipv4.tcp_keepalive_time = 1200
net.ipv4.ip_local_port_range = 10000 65000
net.ipv4.tcp_max_syn_backlog = 8192
net.ipv4.tcp_max_tw_buckets = 5000
net.ipv4.tcp_fastopen = 3
net.ipv4.tcp_mem = 25600 51200 102400
net.ipv4.tcp_rmem = 4096 87380 67108864
net.ipv4.tcp_wmem = 4096 65536 67108864
net.ipv4.tcp_mtu_probing = 1
net.ipv4.tcp_congestion_control = hybla
EOF

sysctl -p               #修改过系统参数后，从/etc/sysctl.conf中加载加载系统参数，一般使用-a
```

和-q两个选项多，-a选项表示显示所有的系统参数

————————————————————脚本2————————————————————

```
#!/usr/bin/env bash
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:~/bin          #设置环境变量
export
PATH
#将环境变量导入系统
#================================================================#
#   System Required:  CentOS 6 or 7                              #
#   Description: Install Shadowsocks-libev server for CentOS 6 or 7 #
#   Author: Teddysun <i@teddysun.com>                            #
#   Thanks: @madeye <https://github.com/madeye>                  #
#   Intro:  https://teddysun.com/357.html                        #
#================================================================#

# Current folder
cur_dir=`pwd`
#将当前地址赋值给cur_dir

libsodium_file="libsodium-1.0.11"                #加密库
libsodium_url="https://github.com/jedisct1/libsodium/releases/download/1.0.11/libsodium-1.0.11.tar.gz"

# Make sure only root can run our script
rootness(){                                      #函数，确保只有root能运行这个脚本
    if [[ $EUID -ne 0 ]]; then                   #有效用户ID不为0
       echo "Error: This script must be run as root!" 1>&2      #将标准输出临时重定向到标准出错，也就是将echo人输出重定向到出错
       exit 1
    fi
}

# Disable selinux
disable_selinux(){                       #关闭selinux
    if [ -s /etc/selinux/config ] && grep 'SELINUX=enforcing' /etc/selinux/config; then
        sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
        setenforce 0       #临时关闭
    fi
}

get_ip(){                #获取ip，ip addr命令类似于ifconfig命令
    local IP=$( ip addr | egrep -o '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | egrep -v "^192\.168|^172\.1[6-9]\.|^172\.2[0-9]\.|^172\.3[0-2]\.|^10\.|^127\.|^255\.|^0\." | head -n 1 )
    [ -z ${IP} ] && IP=$( wget -qO- -t1 -T2 ipv4.icanhazip.com )        #如果IP变量为空，则给IP赋值
    [ -z ${IP} ] && IP=$( wget -qO- -t1 -T2 ipinfo.io/ip )      #如果IP变量还是为空，表示上面的赋值还是不成功，则再赋值
    [ ! -z ${IP} ] && echo ${IP} || echo        #如果IP为非空，则输出IP，如果输出IP失败了，则输出空
}

get_ipv6(){
    local ipv6=$(wget -qO- -t1 -T2 ipv6.icanhazip.com)                          #获取ip6
    if [ -z ${ipv6} ]; then                     #如果ip变量为空
        return 1
```

```
    else                #如果ip变量非空
        return 0
    fi
}

get_char(){                     #stty命令用于显示和修改终端行设置
    SAVEDSTTY=`stty -g`         #-g表示  以stty可读方式打印当前的所有配置
    stty -echo                  #关闭回显。比如在脚本中用于输入密码时
    stty cbreak
    dd if=/dev/tty bs=1 count=1 2> /dev/null    #把指定的输入文件拷贝到指定的输出文件中，并且在拷贝的过程中可以进行格式转换。
    stty -raw
    stty echo                   #打开回显
    stty $SAVEDSTTY
}

#获取最新版本的地址
get_latest_version(){

        #grep打印的是匹配的行，也就是"tag_name": "v3.0.4"，所以后面cut的是该行。cut命令-d用于指定分隔符，所以分隔符是双引号，后面的-f指定显示指定字段的内容
        #所以-f4指第4项内容，也就是v3.0.4，注意，-f0是空的，在第一项引号前面
    ver=$(wget --no-check-certificate -qO- https://api.github.com/repos/shadowsocks/shadowsocks-libev/releases/latest | grep 'tag_name' | cut -d\" -f4)
    [ -z ${ver} ] && echo "Error: Get shadowsocks-libev latest version failed" && exit 1  #如果var为空，则输出内容，退出
    shadowsocks_libev_ver="shadowsocks-libev-$(echo ${ver} | sed -e 's/^[a-zA-Z]//g')"             #组合最新版字符串，括号中的是一组命令
    download_link="https://github.com/shadowsocks/shadowsocks-libev/releases/download/${ver}/${shadowsocks_libev_ver}.tar.gz"
    init_script_link="https://raw.githubusercontent.com/teddysun/shadowsocks_install/master/shadowsocks-libev"
}

#检查是否安装过
check_installed(){
    if [ "$(command -v "$1")" ]; then
        return 0
    else
        return 1
    fi
}

check_version(){
    check_installed "ss-server"
    if [ $? -eq 0 ]; then                       #如果安装过了，则找出版本号
        installed_ver=$(ss-server -h | grep shadowsocks-libev | cut -d' ' -f2)
        get_latest_version                      #获取最新版本的地址
        latest_ver=$(echo ${ver} | sed -e 's/^[a-zA-Z]//g')             #获取最新版本号
        if [ "${latest_ver}" == "${installed_ver}" ]; then              #如果安装的版本是最新版
            return 0
        else
            return 1
        fi
    else
        return 2                        #如果没有安装，则返回2
    fi
}

print_info(){
    clear
```

```
    echo "############################################################"
    echo "# Install Shadowsocks-libev server for CentOS 6 or 7       #"
    echo "# Intro:  https://teddysun.com/357.html                    #"
    echo "# Author: Teddysun <i@teddysun.com>                        #"
    echo "# Github: https://github.com/shadowsocks/shadowsocks-libev #"
    echo "############################################################"
    echo
}

# Check system
check_sys(){
    local checkType=$1
    local value=$2

    local release=''
    local systemPackage=''

    if [[ -f /etc/redhat-release ]]; then
        release="centos"
        systemPackage="yum"
    elif cat /etc/issue | grep -Eqi "debian"; then
        release="debian"
        systemPackage="apt"
    elif cat /etc/issue | grep -Eqi "ubuntu"; then
        release="ubuntu"
        systemPackage="apt"
    elif cat /etc/issue | grep -Eqi "centos|red hat|redhat"; then
        release="centos"
        systemPackage="yum"
    elif cat /proc/version | grep -Eqi "debian"; then
        release="debian"
        systemPackage="apt"
    elif cat /proc/version | grep -Eqi "ubuntu"; then
        release="ubuntu"
        systemPackage="apt"
    elif cat /proc/version | grep -Eqi "centos|red hat|redhat"; then
        release="centos"
        systemPackage="yum"
    fi

    if [[ ${checkType} == "sysRelease" ]]; then                 #判断系统是不是发行版，
注意是双方括号
        if [ "$value" == "$release" ]; then                             #检查版本是否
支持
            return 0
        else
            return 1
        fi
    elif [[ ${checkType} == "packageManager" ]]; then   #检查是不是包管理器
        if [ "$value" == "$systemPackage" ]; then               #检查是哪种包管理器
            return 0
        else
            return 1
        fi
    fi
}

# Get version
getversion(){
    if [[ -s /etc/redhat-release ]]; then                       #如果/etc/redhat-
release非空，这应该是centos系统
        grep -oE  "[0-9.]+" /etc/redhat-release         #打印版本
    else
        grep -oE  "[0-9.]+" /etc/issue
    fi
}
```

```
# CentOS version
centosversion(){
    if check_sys sysRelease centos; then
        local code=$1
        local version="$(getversion)"                                #使用getversion获取版本号，注意getversion是没有返回值的，所以这个实际上就是grep的输出，是字符串，所以用引号
        local main_ver=${version%%.*}                                #version是上面的getversion得到的值，%%.*在Linux命令行与Shell脚本编程大全笔记本最后一部分有解释
        if [ "$main_ver" == "$code" ]; then
            return 0
        else
            return 1
        fi
    else
        return 1
    fi
}

# Pre-installation settings
pre_install(){
    # Check OS system
    if check_sys sysRelease centos; then                             #这里要注意的是函数的调用方式
        # Not support CentOS 5
        if centosversion 5; then
            echo "Not support CentOS 5, please change to CentOS 6 or 7 and try again."
            exit 1
        fi
    else
        echo "Error: Your OS is not supported to run it, please change OS to CentOS and try again."
        exit 1
    fi

    # Check version
    check_version                                                    #返回0，表示是最新版本，返回1，表示不是最新版本
    status=$?                                                        #$?: 上个命令的退出状态，或函数的返回值
    if [ ${status} -eq 0 ]; then
        echo "Latest version ${shadowsocks_libev_ver} has been installed, nothing to do..."
        echo
        exit 0
    elif [ ${status} -eq 1 ]; then
        echo "Installed version: ${installed_ver}"
        echo "Latest version: ${latest_ver}"
        echo "Upgrade shadowsocks libev to latest version..."
        ps -ef | grep -v grep | grep -i "ss-server" > /dev/null 2>&1    #如果不是最新版本，则停止进程
        if [ $? -eq 0 ]; then
            /etc/init.d/shadowsocks stop                             #停止进程
        fi
    elif [ ${status} -eq 2 ]; then                                   #系统中没有安装ssh时，status为2
        print_info
        get_latest_version
        echo "Latest version: ${shadowsocks_libev_ver}"
        echo
    fi

    # Set shadowsocks-libev config password
    echo "Please input password for shadowsocks-libev"
    read -p "(Default password: teddysun.com):" shadowsockspwd       #将输入放到shadowsockspwd变量，参数-p表示有提示语句
```

```
    [ -z "${shadowsockspwd}" ] && shadowsockspwd="teddysun.com"          #如果变量为空，则将变量赋值为默认变量
    echo
    echo "---------------------------"
    echo "password = ${shadowsockspwd}"
    echo "---------------------------"
    echo

    # Set shadowsocks-libev config port
    while true
    do
    echo -e "Please input port for shadowsocks-libev [1-65535]"          #-e表示打开反斜杠Esc转义
    read -p "(Default port: 8989):" shadowsocksport
    [ -z "$shadowsocksport" ] && shadowsocksport="8989"
    expr ${shadowsocksport} + 0 &>/dev/null                      #将端口号+0，并永久重定向到/dev/null，这个加0应该是转为数字
    if [ $? -eq 0 ]; then
        if [ ${shadowsocksport} -ge 1 ] && [ ${shadowsocksport} -le 65535 ]; then    #判断端口号是否符合要求
            echo
            echo "---------------------------"
            echo "port = ${shadowsocksport}"
            echo "---------------------------"
            echo
            break                                           #出错，则跳出这次循环，这里应该指不再执行下面这个echo
        else
            echo "Input error, please input correct number"
        fi
    else
        echo "Input error, please input correct number"
    fi
    done

    echo
    echo "Press any key to start...or press Ctrl+C to cancel"
    char=`get_char`                                         #反引号`允许你将shell命令的输出赋值给变量
    #Install necessary dependencies
    yum install -y epel-release
    yum install -y gcc gettext-devel unzip autoconf automake make zlib-devel libtool xmlto asciidoc udns-devel libev-devel
    yum install -y pcre pcre-devel perl perl-devel cpio expat-devel openssl-devel mbedtls-devel
}

# Download latest shadowsocks-libev
download_files(){
    cd ${cur_dir}                                   #进入目录

    if ! wget --no-check-certificate -O ${shadowsocks_libev_ver}.tar.gz ${download_link}; then           #获取文件
        echo "Failed to download ${shadowsocks_libev_ver}.tar.gz"
        exit 1
    fi

    if ! wget --no-check-certificate -O ${libsodium_file}.tar.gz ${libsodium_url}; then
        echo "Failed to download ${libsodium_file}.tar.gz"
        exit 1
    fi

    # Download init script
    if ! wget --no-check-certificate -O /etc/init.d/shadowsocks ${init_script_link}; then
```

```
        echo "Failed to download shadowsocks-libev init script!"
        exit 1
    fi
}

# Config shadowsocks            配置shadowsocks
config_shadowsocks()
{
    local server_value="\"0.0.0.0\""            #本身内容是有双引号的，所以这里有4个双引号
    if get_ipv6; then
        server_value="[\"[::0]\",\"0.0.0.0\"]"
    fi

    if [ ! -d /etc/shadowsocks-libev ]; then
        mkdir -p /etc/shadowsocks-libev
    fi
    cat > /etc/shadowsocks-libev/config.json<<-EOF                      #将配置写入文
件
{
    "server":${server_value},
    "server_port":${shadowsocksport},
    "local_address":"127.0.0.1",
    "local_port":1080,
    "password":"${shadowsockspwd}",
    "timeout":600,
    "method":"aes-256-cfb"
}
EOF
}

# Firewall set                  设置防火墙
firewall_set(){
    echo "firewall set start..."
    if centosversion 6; then
        /etc/init.d/iptables status > /dev/null 2>&1
        if [ $? -eq 0 ]; then
            iptables -L -n | grep -i ${shadowsocksport} > /dev/null 2>&1        #-l
列出所有规则，-n以数字形式列出规则
            if [ $? -ne 0 ];
then                                                    #注意这个是不等于
                iptables -I INPUT -m state --state NEW -m tcp -p tcp --dport $
{shadowsocksport} -j ACCEPT
                iptables -I INPUT -m state --state NEW -m udp -p udp --dport $
{shadowsocksport} -j ACCEPT
                /etc/init.d/iptables save
                /etc/init.d/iptables restart
            else
                echo "port ${shadowsocksport} has been set up."
            fi
        else
            echo "WARNING: iptables looks like shutdown or not installed, please
manually set it if necessary."
        fi
    elif centosversion 7; then
        systemctl status firewalld > /dev/null 2>&1
        if [ $? -eq 0 ]; then
            firewall-cmd --permanent --zone=public --add-port=${shadowsocksport}/
tcp
            firewall-cmd --permanent --zone=public --add-port=${shadowsocksport}/
udp
            firewall-cmd --reload
        else
            echo "Firewalld looks like not running, try to start..."
            systemctl start firewalld
            if [ $? -eq 0 ]; then
                firewall-cmd --permanent --zone=public --add-port=$
{shadowsocksport}/tcp
```

```
                firewall-cmd --permanent --zone=public --add-port=${shadowsocksport}/udp
                firewall-cmd --reload
            else
                echo "WARNING: Try to start firewalld failed. please enable port ${shadowsocksport} manually if necessary."
            fi
        fi
    fi
    echo "firewall set completed..."
}

# Install Shadowsocks-libev
install_shadowsocks(){
    if [ ! -f /usr/lib/libsodium.a ]; then              #如果不存在这个动态链接库，这个库是加密库
        cd ${cur_dir}
#进入目录
        tar zxf ${libsodium_file}.tar.gz
        cd ${libsodium_file}
        ./configure --prefix=/usr && make && make install               #配置、编译、安装
        if [ $? -ne 0 ]; then
            echo "${libsodium_file} install failed!"
            exit 1
        fi
    fi

    ldconfig                    #这个命令是为了将这个加密库给系统共享
    cd ${cur_dir}
    tar zxf ${shadowsocks_libev_ver}.tar.gz     #解压安装包
    cd ${shadowsocks_libev_ver}
    ./configure
    make && make install
    if [ $? -eq 0 ]; then
        chmod +x /etc/init.d/shadowsocks
        # Add run on system start up
        chkconfig --add shadowsocks             #添加为系统
        chkconfig shadowsocks on                #开机自启动
        # Start shadowsocks
        /etc/init.d/shadowsocks start           #启动
        if [ $? -eq 0 ]; then
            echo "Shadowsocks-libev start success!"
        else
            echo "Shadowsocks-libev start failure!"
        fi
    else
        echo
        echo "Shadowsocks-libev install failed! Please visit https://teddysun.com/357.html and contact."
        exit 1
    fi

    cd ${cur_dir}
    rm -rf ${shadowsocks_libev_ver} ${shadowsocks_libev_ver}.tar.gz
    rm -rf ${libsodium_file} ${libsodium_file}.tar.gz

    clear
    echo
    echo "Congratulations, Shadowsocks-libev install completed!"
    echo -e "Your Server IP: \033[41;37m $(get_ip) \033[0m"
    echo -e "Your Server Port: \033[41;37m ${shadowsocksport} \033[0m"
    echo -e "Your Password: \033[41;37m ${shadowsockspwd} \033[0m"
    echo -e "Your Local IP: \033[41;37m 127.0.0.1 \033[0m"
    echo -e "Your Local Port: \033[41;37m 1080 \033[0m"
```

```
    echo -e "Your Encryption Method: \033[41;37m aes-256-cfb \033[0m"
    echo
    echo "Welcome to visit:https://teddysun.com/357.html"
    echo "Enjoy it!"
    echo
}

# Uninstall Shadowsocks-libev
uninstall_shadowsocks_libev(){
    print_info
    printf "Are you sure uninstall shadowsocks-libev? (y/n)"
    printf "\n"
    read -p "(Default: n):" answer
    [ -z ${answer} ] && answer="n"      #如果选择是n，则什么也不做

    if [ "${answer}" == "y" ] || [ "${answer}" == "Y" ]; then
        ps -ef | grep -v grep | grep -i "ss-server" > /dev/null 2>&1    #则查找ss-
server进程
        if [ $? -eq 0 ]; then                    #找到进程则停止进程
            /etc/init.d/shadowsocks stop
        fi
        chkconfig --del shadowsocks     #删除服务
        rm -fr /etc/shadowsocks-libev
        rm -f /usr/local/bin/ss-local
        rm -f /usr/local/bin/ss-tunnel
        rm -f /usr/local/bin/ss-server
        rm -f /usr/local/bin/ss-manager
        rm -f /usr/local/bin/ss-redir
        rm -f /usr/local/bin/ss-nat
        rm -f /usr/local/lib/libshadowsocks-libev.a
        rm -f /usr/local/lib/libshadowsocks-libev.la
        rm -f /usr/local/include/shadowsocks.h
        rm -f /usr/local/lib/pkgconfig/shadowsocks-libev.pc
        rm -f /usr/local/share/man/man1/ss-local.1
        rm -f /usr/local/share/man/man1/ss-tunnel.1
        rm -f /usr/local/share/man/man1/ss-server.1
        rm -f /usr/local/share/man/man1/ss-manager.1
        rm -f /usr/local/share/man/man1/ss-redir.1
        rm -f /usr/local/share/man/man1/ss-nat.1
        rm -f /usr/local/share/man/man8/shadowsocks-libev.8
        rm -fr /usr/local/share/doc/shadowsocks-libev
        rm -f /etc/init.d/shadowsocks
        echo "Shadowsocks-libev uninstall success!"
    else
        echo
        echo "uninstall cancelled, nothing to do..."
        echo
    fi
}

# Install Shadowsocks-libev
install_shadowsocks_libev(){            #安装shadowsocks，这里调用了一堆函数
    rootness                    #判断是不是root用户
    disable_selinux             #关闭selinux
    pre_install                 #安装前的准备工具，包括选择好端口和安装各种依赖包
    download_files              #下载文件
    config_shadowsocks          #配置shadowsocks，事实上就是新增一个config.json文件
    firewall_set                #设置防火墙
    install_shadowsocks         #安装
}

# Initialization step
action=$1
[ -z $1 ] && action=install     #如果参数为空，则将动作设置为安装
case "$action" in
```

```
    install|uninstall)
        ${action}_shadowsocks_libev
        ;;
    *)
        echo "Arguments error! [${action}]"
        echo "Usage: `basename $0` [install|uninstall]"
        ;;
esac
```