

Setting Up a ShadowsocksR Proxy Server

 bmao.tech/blog/shadowsocksr



101对双生儿

Not pleased by external gains. Not saddened by personal losses.

2018-06-04

Follow this tutorial to start setting up a ShadowsocksR proxy server and enjoying freedom outside the wall!

Server-side Setup

Prepare a server

I would recommend [DigitalOcean](#) and [Vultr](#), both of which offer various plans for [virtual private servers \(VPS\)](#). DigitalOcean provides [\\$100 credits](#) for new users, while Vultr provides [\\$25 credits](#) for new users. Students with a valid .edu email address can sign up for [GitHub Student Developer Pack](#) and get an extra \$50 credits on DigitalOcean. [Google Cloud](#) and [Microsoft Azure](#), at the same time, also provide free credits for trial, but they can exhaust faster. These free credits usually last for only one year, so you should take full advantage of them before they expire. If you have access to a dedicated server, keep in mind that you should have full access to it.

I will deploy a Vultr Cloud Compute (VC2) server with Ubuntu 18.04 in this tutorial.

```
1. root@vultr-ubuntu-nj1: ~ (ssh)
(django) Steven@Stevens-MacBook-Pro ~/Documents/projects/blog $ ssh -X root@
The authenticity of host ' ' can't be established.
ECDSA key fingerprint is .
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added ' ' (ECDSA) to the list of known hosts.
root@ 's password:
Warning: untrusted X11 forwarding setup failed: xauth key data not generated
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-20-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon May 28 00:30:54 UTC 2018

System load:  0.27           Processes:            94
Usage of /:   8.3% of 19.63GB Users logged in:       0
Memory usage: 24%           IP address for ens3:
Swap usage:   0%

30 packages can be updated.
15 updates are security updates.

root@vultr-ubuntu-nj1:~#
```

Install ShadowsocksR Proxy Server

In this tutorial, we will use [a script](#) developed by [@Toyo](#) to help us install the proxy server. This script installs and runs the proxy server in the MudbJSON mode so that we can manage each user of the proxy server separately. For those who are interested in the technical details of this script, please refer to [『原创』ShadowsocksR MudbJSON模式多用户一键脚本 支持流量限制等](#) (Chinese).

Thanks to the script, the installation can be extremely easy. First, download and run the script.

```
1 wget -N --no-check-certificate
  https://raw.githubusercontent.com/ToyoDAdoubi/doubi/master/ssrmu.sh && chmod +x ssrmu.sh &&
  ./ssrmu.sh
```

```
1. root@vultr-ubuntu-nj1: ~ (ssh)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Mon May 28 00:30:54 UTC 2018

System load: 0.27          Processes:           94
Usage of /:  8.3% of 19.63GB Users logged in:      0
Memory usage: 24%          IP address for ens3: 
Swap usage:  0%

30 packages can be updated.
15 updates are security updates.

root@vultr-ubuntu-nj1:~# wget -N --no-check-certificate https://softs.loan/Bash/ssrmu.sh && chmod +x ssrmu.sh && bash ssrmu.sh
--2018-05-28 00:31:25-- https://softs.loan/Bash/ssrmu.sh
Resolving softs.loan (softs.loan)... 168.235.104.198
Connecting to softs.loan (softs.loan)|168.235.104.198|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 79473 (78K) [application/octet-stream]
Saving to: 'ssrmu.sh'

ssrmu.sh          100%[=====>]  77.61K  460KB/s   in 0.2s

2018-05-28 00:31:26 (460 KB/s) - 'ssrmu.sh' saved [79473/79473]
```

Enter **1** to start the installation of ShadowsocksR proxy server.

```
1. root@vultr-ubuntu-nj1: ~ (ssh)

2018-05-28 00:31:26 (460 KB/s) - 'ssrmu.sh' saved [79473/79473]

ShadowsocksR MuJSON一键管理脚本 [v1.0.25]
---- Toyo | doub.io/ss-jc60 ----

1. 安装 ShadowsocksR
2. 更新 ShadowsocksR
3. 卸载 ShadowsocksR
4. 安装 libsodium(chacha20)

-----
5. 查看 账号信息
6. 显示 连接信息
7. 设置 用户配置
8. 手动 修改配置
9. 配置 流量清零

-----
10. 启动 ShadowsocksR
11. 停止 ShadowsocksR
12. 重启 ShadowsocksR
13. 查看 ShadowsocksR 日志

-----
14. 其他功能
15. 升级脚本

当前状态: 未安装

请输入数字 [1-15]: 1
```

First, hit the **Enter** key to let the script automatically detect the IP address of the server. Then enter **user name**, **port number** and **password** for the first user.

The username and password should only contain ASCII characters. The port entered here should not have conflict with the ports which the server is using or the server system reserves.

```
1. root@vultr-ubuntu-nj1: ~ (ssh)

当前状态： 未安装

请输入数字 [1-15]: 1
[信息] 开始设置 ShadowsocksR账号配置...
请输入用户配置中要显示的 服务器IP或域名 (当服务器有多个IP时, 可以指定用户配置中显示的IP或者域名)
(默认自动检测外网IP):

-----
IP或域名 : 192.168.1.1
-----

请输入要设置的用户 用户名(请勿重复, 用于区分, 不支持中文, 会报错 ! )
(默认: doubi):steven

-----
用户名 : steven
-----

请输入要设置的用户 端口(请勿重复, 用于区分)
(默认: 2333):9000

-----
端口 : 9000
-----

请输入要设置的用户 密码
(默认: doub.io):
```

For the encryption method, select **"none"** (i.e. enter **1**) since I will use **"auth_chain_a"** protocol later.

```
1. root@vultr-ubuntu-nj1: ~ (ssh)

请选择要设置的用户 加密方式
1. none
[注意] 如果使用 auth_chain_* 系列协议, 建议加密方式选择 none (该系列协议自带 RC
4 加密), 混淆随意

2. rc4
3. rc4-md5
4. rc4-md5-6

5. aes-128-ctr
6. aes-192-ctr
7. aes-256-ctr

8. aes-128-cfb
9. aes-192-cfb
10. aes-256-cfb

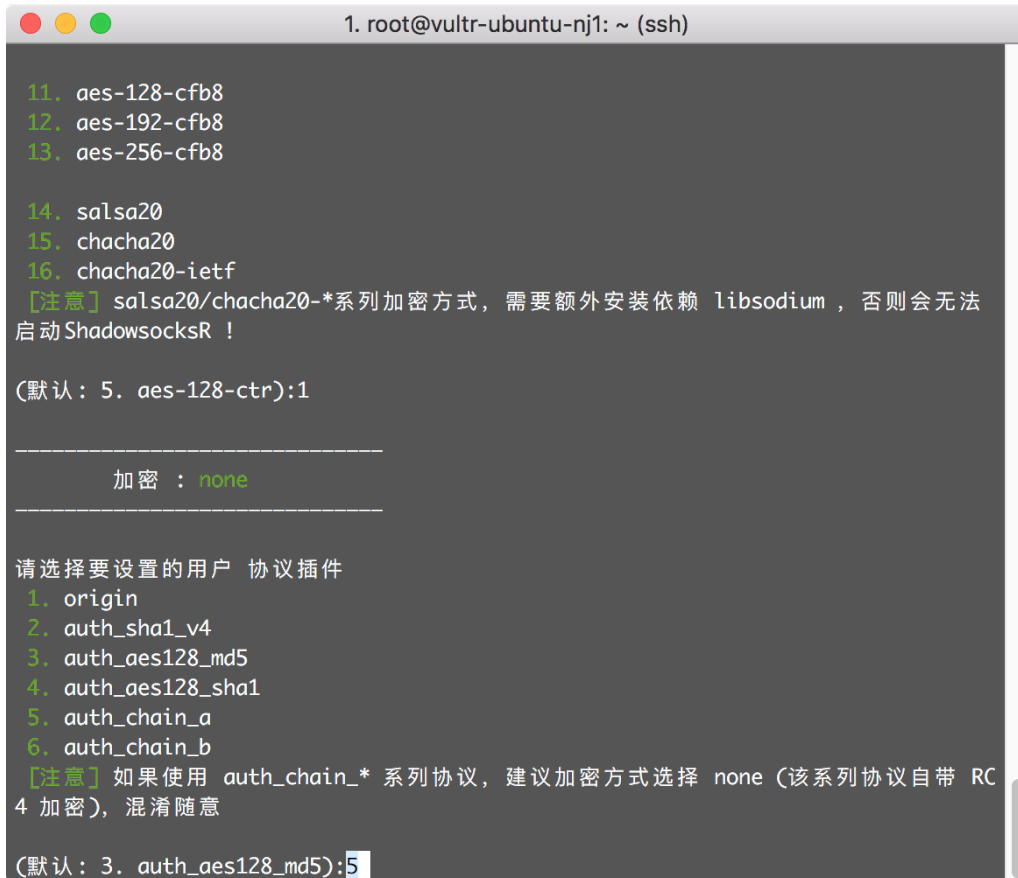
11. aes-128-cfb8
12. aes-192-cfb8
13. aes-256-cfb8

14. salsa20
15. chacha20
16. chacha20-ietf
[注意] salsa20/chacha20-*系列加密方式, 需要额外安装依赖 libsodium , 否则会无法
启动ShadowsocksR !

(默认: 5. aes-128-ctr):1
```

Now select **"auth_chain_a"** protocol (i.e. enter **5**) because it is safer.

For more information about “auth_chain_a” protocol, please refer to [协议auth_chain_a是啥](#) (Chinese) or https://github.com/shadowsocksr-backup/shadowsocks-rss/blob/master/doc/auth_chain_a.md (English). You might also use “auth_chain_b” protocol, but it has not been widely supported on the iOS devices yet.



```
1. root@vultr-ubuntu-nj1: ~ (ssh)

11. aes-128-cfb8
12. aes-192-cfb8
13. aes-256-cfb8

14. salsa20
15. chacha20
16. chacha20-ietf
[注意] salsa20/chacha20-*系列加密方式, 需要额外安装依赖 libsodium, 否则会无法启动ShadowsocksR !

(默认: 5. aes-128-ctr):1

-----
加密 : none
-----

请选择要设置的用户 协议插件
1. origin
2. auth_sha1_v4
3. auth_aes128_md5
4. auth_aes128_sha1
5. auth_chain_a
6. auth_chain_b
[注意] 如果使用 auth_chain_* 系列协议, 建议加密方式选择 none (该系列协议自带 RC4 加密), 混淆随意

(默认: 3. auth_aes128_md5):5
```

Use the **default setting** for obfuscation (i.e. enter **5**) with no compatibility with Shadowsocks (i.e. enter **n**). Original Shadowsocks proxy is no longer safe, so it is not necessary to make the server compatible with Shadowsocks.

For more technical information about encryption method, protocol and obfuscation in ShadowsocksR, please refer to [ShadowsocksR 协议插件文档](#) (Chinese).

```
1. root@vultr-ubuntu-nj1: ~ (ssh)

6. auth_chain_b
[注意] 如果使用 auth_chain_* 系列协议, 建议加密方式选择 none (该系列协议自带 RC4 加密), 混淆随意

(默认: 3. auth_aes128_md5):5

-----
协议 : auth_chain_a
-----

请选择要设置的用户 混淆插件
1. plain
2. http_simple
3. http_post
4. random_head
5. tls1.2_ticket_auth
[注意] 如果使用 ShadowsocksR 代理游戏, 建议选择 混淆兼容原版或 plain 混淆, 然后客户端选择 plain, 否则会增加延迟 !
另外, 如果你选择了 tls1.2_ticket_auth, 那么客户端可以选择 tls1.2_ticket_fastauth, 这样即能伪装特征 又不会增加延迟 !

(默认: 5. tls1.2_ticket_auth):

-----
混淆 : tls1.2_ticket_auth
-----

是否设置 混淆插件兼容原版(_compatible)? [Y/n]n
```

Use the **default settings** for all remaining parameters unless you want to limit the device number, the network speed, the traffic and/or the ports of the current user.

```
1. root@vultr-ubuntu-nj1: ~ (ssh)

是否设置 混淆插件兼容原版(_compatible)? [Y/n]n

请输入要设置的用户 欲限制的设备数 ( auth_* 系列协议 不兼容原版才有效 )
[注意] 设备数限制: 每个端口同一时间能链接的客户端数量(多端口模式, 每个端口都是独立计算), 建议最少 2个。
(默认: 无限):

请输入要设置的用户 单线程 限速上限(单位: KB/S)
[注意] 单线程限速: 每个端口 单线程的限速上限, 多线程即无效。
(默认: 无限):

请输入要设置的用户 总速度 限速上限(单位: KB/S)
[注意] 端口总限速: 每个端口 总速度 限速上限, 单个端口整体限速。
(默认: 无限):

请输入要设置的用户 可使用的总流量上限(单位: GB, 1-838868 GB)
(默认: 无限):

请输入要设置的用户 禁止访问的端口
[注意] 禁止的端口: 例如不允许访问 25端口, 用户就无法通过SSR代理访问 邮件端口25了, 如果禁止了 80,443 那么用户将无法访问 http/https 网站。
封禁单个端口格式: 25
封禁多个端口格式: 23,465
封禁 端口段格式: 233-266
封禁多种格式端口: 25,465,233-666 (不带冒号:)
(默认为空 不禁止访问任何端口):
```

Finally, the script shows that the proxy server is installed successfully, and the first user is configured. Write down the ShadowsocksR URL starting with `ssr://` or the QR code URL so that we can add this configuration to the clients later.

```
1. root@vultr-ubuntu-nj1: ~ (ssh)
[信息] ShadowsocksR 启动成功 !

=====

用户 [steven] 的配置信息:

I P      : 149.28.52.24
端口     : 9000
密码     : 
加密     : none
协议     : auth_chain_a
混淆     : tls1.2_ticket_auth
设备数限制 : 0(无限)
单线程限速 : 0 KB/S
用户总限速 : 0 KB/S
禁止的端口 : 无限制

已使用流量 : 上传: 0 B + 下载: 0 B = 0 B
剩余的流量 : 819.21 TB
用户总流量 : 819.21 TB

SSR 链接 : 
SSR 二维码 : http://doub.pw/qr/qr.php?text=

提示:
在浏览器中, 打开二维码链接, 就可以看到二维码图片。
协议和混淆后面的[ _compatible ], 指的是 兼容原版协议/混淆。
```

If you want to add another user, just run the script again by `./ssrmu.sh`, choose **7** and follow the exact same instruction.

In theory, we should boost our proxy server's TCP connection. The good news is that Ubuntu 18.04 enables Google BBR by default, so we do not need to take any further action. If you use other versions or systems and need to enable it manually, please follow this tutorial: Centos/Ubuntu/Debian BBR加速一键安装包 (Chinese).

Debian 9 users can use the following commands to enable Google BBR:

```
1 echo "net.core.default_qdisc=fq" >> /etc/sysctl.conf
2 echo "net.ipv4.tcp_congestion_control=bbr" >>
  /etc/sysctl.conf
3 sysctl -p
```

For more information, please check out this post: Debian 9快速开启Google BBR的方法, 实现高效单边加速.

And don't forget to set firewall rules for the server! DigitalOcean and Vultr provide graphical interface inside their consoles so that users can set up firewalls in clicks.

Client-side Setup

I would recommend the following client-side applications:

Reference

1. Toyo, "『原创』ShadowsocksR MudbJSON模式多用户一键脚本 支持流量限制等," 逗

比根据地. [Online]. Available: <https://doub.io/ss-jc60/>.

Deploying OneIndex on Heroku

持美国 F-1 签证在纽约市面签加拿大临时居民访问签证

Contents

2018 101对双生儿

Hexo Theme Yelee by MOxFIVE