



ShadowsocksR PC客户端中的 [代理规则 – 用户自定义] 功能使用教程

2017-11-14 | [higeon](#) | [技术](#)
by Toyo

[逗比根据地 \(https://doub.io/\)](https://doub.io/) / 2017-11-12 09:57

文章目录

[\[显示\]](#)

本文最后更新于 **2017年11月12日 08:56** 可能会因为没有更新而失效。如已失效或需要修正，请留言！ShadowsocksR PC客户端在 4.6.0 版本中添加了一个代理规则选项 [用户自定义]，说实话我一直没注意到，还是一个逗比在网站问我这个功能的作用是什么时，我才突然发现这个功能，就特地去了解一下，然后写个简单的使用教程。

首先先明白 代理规则 是干什么的，和PAC（系统代理模式）是什么区别：

[点击展开 查看更多 \(https://doub.io/ss-jc61/#\)](https://doub.io/ss-jc61/#)

ShadowsocksR目前分两种规则方式，那就是 系统代理模式（PAC、全局，主要解释PAC）和 代理规则。

PAC规则 是根据PAC文件中的黑白地址名单来判断那些网站走代理。也就是判断 流量是否进入客户端。

当你要访问的网站满足 PAC里的设置，那么浏览器就会向代理服务器也就是客户端发送网站请求数据，这时候我们才进入了ShadowsocksR客户端，接下来我们就进入了代理规则判断环节。

代理规则 是根据IP判断，按选择的规则来判断进入 客户端的流量是直连还是走代理。

当你访问 XXX 网站，然后是全局或者满足PAC条件（不满足条件就直接直连访问了，也不会有接下来的代理规则判断了）从而访问 **XXX**网站的请求数据流量进入了客户端，然后客户端会根据 **XXX**网站的IP来判断，如下：

- 绕过局域网：当IP属于局域网内的，那么SSR客户端就会让流量直连，反之则会让流量走代理（发送到SSR服务端）。
- 绕过局域网和大陆：当IP属于大陆内或局域网的，那么SSR客户端就会让流量直连，反之则会让流量走代理（发送到SSR服务端）。
- 绕过局域网和非大陆：当IP属于大陆外(非大陆IP都算大陆外)或局域网的，那么SSR客户端就会让流量直连，反之则会让流量走代理（发送到SSR服务端）。

- 全局：无论什么情况都直接走代理。

这两种判断方式是相互配合使用的，先用系统代理模式来判断是否让数据进入SSR客户端，再用代理规则来判断进入SSR客户端的数据是直连还是走代理。

举个栗子：假设系统代理模式为PAC，那么访问 [www.google.com \(http://www.google.com\)](http://www.google.com)，浏览器在PAC文件中匹配这个域名，并发现这个域名按PAC规则应走代理，所以浏览器就会发送访问网页数据到PAC中的代理服务器(默认如127.0.0.1:1080)，于是SSR客户端就收到了访问谷歌的数据，而这时候就该用代理规则判断了。

代理规则为：绕过局域网，则判断 [www.google.com \(http://www.google.com\)](http://www.google.com) 域名的IP是否是局域网IP，然而不是局域网IP，于是走代理。

代理规则为：绕过局域网和大陆，则判断 [www.google.com \(http://www.google.com\)](http://www.google.com) 域名的IP是否是局域网IP 或 大陆IP，然而不是局域网IP或大陆IP，于是走代理。

代理规则为：绕过局域网和非大陆，则判断 [www.google.com \(http://www.google.com\)](http://www.google.com) 域名的IP是否是局域网IP 或 非大陆IP，然而不是非大陆IP，于是不走代理，直连。

代理规则为：全局，不判断 [www.google.com \(http://www.google.com\)](http://www.google.com) 域名的IP，直接走代理。

你们大概能明白了吗？不明白可以评论区问我。

代理规则中的 用户自定义，就是可以让你自定义什么域名或IP 走代理还是直连，比PAC更方便添加/编辑规则，以后完全可以抛弃 PAC，而改用 用户自定义 代理规则（系统代理模式选择全局，浏览器扩展也设置 127.0.0.1 1080(默认端口)）。

实际上 SSR 安卓客户端中的 ACL 和 代理规则 - 用户自定义 是一样的。

接下来讲解代理规则格式等内容。

文件位置

代理规则 - 用户自定义 的规则文件是 ShadowsocksR.exe 客户端文件同目录下的 **user.rule** 文件（如果没有自己新建）。

注意：每次修改规则文件后，都需要 重启SSR客户端 才能应用最新规则。

提供一个Github上面找到的 ACL（PC客户端 规则文件，带去广告）：

<https://raw.githubusercontent.com/ACL4SSR/ACL4SSR/master/gfwlist-user.rule>
(<https://raw.githubusercontent.com/ACL4SSR/ACL4SSR/master/gfwlist-user.rule>)

如果不满意，可以去 **Github** 搜索 **ACL**，应该会搜到类似的文件！

你们可以边参考这个规则，边理解下面的解释，当然最好自己动手多测试几次。

当然你也可以直接下载这个现成的规则文件到ShadowsocksR客户端同目录下，并重命名为 **user.rule** 文件名拿来用。

文件格式

注释

以#号开始的一行为注释内容，注意：#号的前面不得有空格，例如：

```
# 我是注释，我会被忽略，第一行首字符必须是#号才行。
```

基本规则

规则文件内，除了空行和注释，其它的每行都是一条规则，规则之间有先后次序之分。若出现相同的规则，那么后一条规则可覆盖前一条规则。

规则分两类：

规则匹配结果类型

规则匹配结果类型有四种：**remoteproxy**、**localproxy**、**direct**、**reject**。均可用于两类规则。

四种结果的含义：

remoteproxy：经过SSR服务器连接（走代理） **localproxy**：经过本地代理连接，或没有配置本

域名规则

格式： `hostname rule # 域名 规则`

hostname 格式支持三种写法：

直接写完整域名，如 [b.com \(http://b.com\)](http://b.com)，此时不匹配 [a.b.com \(http://a.b.com\)](http://a.b.com)。这类 [b.com \(http://b.com\)](http://b.com) 将继续往下面判断，如果下面没有 [b.com \(http://b.com\)](http://b.com) 的规则，那

注意：[b.com \(http://b.com\)](http://b.com) 与 [.b.com \(http://b.com\)](http://b.com) 是会相互覆盖的（后写的有效），

域名规则还有一种特殊的规则 rule 写法，例如（不过一般用不上，忽略即可）：

```
a.com \(http://a.com\). 127.0.0.1  .b.com \(http://b.com\). 123.123.123.123
```

这种规则相当于给指定域名 ([a.com \(http://a.com\)](http://a.com)) 或一组子域名 ([.b.com \(http://b.com\)](http://b.com)) 一个相应的IP，但不直接指定最终结果。

最终结果将由IP段规则决定，比如接下来的IP规则判断中，**127.0.0.1** 规则判断结果是直连，那么将使用此规则所指定的IP来连接。

IP段规则

格式： ip1 ip2 rule # IP段起始 IP段结尾 规则 # 如果只要控制一个IP，那么 ip1 和

注意：ip2 必须大于或等于 ip1，此规则应用于包含 ip1、ip2 及它们之间的所有IP。如果两

利用这个规则可以编写基于黑名单或白名单的IP规则，例如在规则文件内先写：

```
:: ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff remoteproxy 0.0.0.0 255.255.255.255
```

前两行这样意味着所有的IP（第一行是IPv6全部IP段，第二行是IPv4全部IP段）默认走代理，然后利用后一规则覆盖前一规则来设置不走代理的IP，所以这就是白名单规则。

反过来如果所有IP设置为 **localproxy**，那么就是黑名单规则：

```
:: ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff localproxy 0.0.0.0 255.255.255.255
```

规则匹配次序

当你访问 www.google.com (<http://www.google.com>)，浏览器配置全局模式（或者PAC默认，但是谷歌被墙，PAC判断走代理），然后浏览器会把谷歌访问请求发给SSR客户端，然后：

▣ SSR客户端代理规则选择：用户自定义（你已经配置好了规则），然后SSR客户端会去代理规则

全网也只有SSR的WIKI才有这个教程，所以我参考WIKI的内容，因为原版WIKI就很详细了，所以我只是在原版内容的基础上做了一点补充。

因为SSR项目已删除，所以我就引用了SSRR fork的WIKI：

<https://github.com/shadowsocksrr/shadowsocks-rss/wiki/C%23-Proxy-Rule>
(<https://github.com/shadowsocksrr/shadowsocks-rss/wiki/C%23-Proxy-Rule>)

转载请超链接注明：[逗比根据地 \(https://doub.io\)](https://doub.io) » [ShadowsocksR PC客户端中的 \[代理规](#)

责任声明：本站一切资源仅用作交流学习，请勿用作商业或违法行为！如造成任何后果，本站概不

[赞\(2\)or打赏 \(https://doub.io/dashang/\)](#)

Shared via [Inoreader \(http://www.inoreader.com\)](http://www.inoreader.com)

□

© 作者 [WORDPRESS.COM](http://www.wordpress.com).