v2ray 代理的使用

内部文件，请勿外传！

一、有关正向代理的原理请查看 gt-it / **CommonDocuments** 里的《网络代理技术原理.pdf》，以及配置 windows10 使用正向代理上网请查看 gt-it / **CommonDocuments** 里的《使用正向代理上网.pdf》,若要自己生成 ssl 证书，请查看 gt-it / **CommonDocuments** 里的《SSL 证书原理及格式 1.2.pdf》

二、由于 GFW 的封锁，常规的代理及 vpn 技术无法过境，所以推荐使用 v2ray 的 websocket 伪装技术。

简单的说就是把要走代理通道的流量放在 https 流量里，从外表上看就是进行正常的 https 的交互，https 从第三方来看只能看到访问的目标 ip:port 及 sni 里的域名。所以只要 ip:port:sni 任一项不在封锁的范围内，就能顺利通过 GFW。

为了使伪装得像点样子，我们需要使用真正的 web 服务器做为入口，然后做个次级路径的分流，比如访问 /vtest 路径的 https 流量 分流到 v2ray 服务器，然后 v2ray 验证客户端的相关 id，验证通过后再进行代理的操作，代理得到的结果再通过 web 服务器发回给客户端。（所以在代理服务端要运行 2 个服务进程，一个是 web 服务进程，一个是 v2ray 进程）

三、nginx 的配置

在 TC 云的 GZ 区的 42.194.213.182 这台虚拟机上我们运行的是 nginx，所以就用 nginx 做为 web 服务器，在上面开一个 server，监听 vtest.cdn.tencent.com 域名，配置如下（/etc/nginx/nginx.conf）

```
http {
……此处略去若干行！
#####  the server below is v2ray proxy's stealthy web host , date: 2020-07-16  #####
    server {
        listen 9833 ssl;     #监听 9833 ssl 端口，防火墙要开放 9833 端口给相关人员访问
        server_name vtest.cdn.tencent.com;  #此域名为虚构的，尽量用国内知名的，不易被屏蔽
        ssl_certificate /etc/v2ray/vtest.cdn.tencent.com.crt;    #证书及 key 文件可自己创建
        ssl_certificate_key  /etc/v2ray/vtest.cdn.tencent.com.key;
        ssl_session_cache shared:SSL:1m;
        ssl_session_timeout 10m;
        ssl_ciphers HIGH:!aNULL:!MD5;
        ssl_prefer_server_ciphers on;
        location /vtest {                    #必须和 v2ray 配置里的"path"一致
                proxy_redirect off;
                proxy_pass http://127.0.0.1:3858;    #把访问/vtest 路径的分流到 3858 端口
                proxy_http_version 1.1;      #这个本地 3858 端口为 v2ray 进程监听
                proxy_set_header Upgrade $http_upgrade;
```

```
                proxy_set_header Connection "upgrade";
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_intercept_errors on;
                }
        location / {
                root /vtest;  #服务器的/vtest 目录作为 vtest.cdn.tencent.com 站点的根目录
                index index.html; #当 GFW 做主动监测时，会访问此站点，所以要放几个正常的
            }                       # html 文件，以便让它相信这是一个网站
        }#  // end of server
}  #// end of http
```

如果后续要修改伪装的域名，服务端只需修改 nginx.conf 里的 server_name 及 ssl_certificate 和 key 文件，然后生成新的 key 及 crt 文件放到/etc/v2ray/目录下，

再 $ /usr/sbin/nginx -t 检查配置是否正确

$ /usr/sbin/nginx -s reload 重加载 nginx 配置即可，最后在客户端修改伪装的域名就行了。

四、v2ray 服务端的安装及配置（在 TC 云的 GZ 区的 42.194.213.182 这台虚拟机上）

（4.1）不推荐使用一键安装脚本，我们使用手动部署的方式

首先在 https://github.com/v2ray/v2ray-core/releases 这里查看 v2ray 版本，选择要下载的版本，centos 64 位的就下载 v2ray-linux-64.zip 这个压缩文档，（可以先在浏览器上点击要下载的文档，再复制下载链接）操作如下：

```
root@tc# mkdir /root/v2ray      #创建/root/v2ray 目录，
root@tc# cd /root/v2ray      #进入/root/v2ray 目录，
root@tc# wget https://github.com/v2ray/v2ray-core/releases/download/v4.26.0/v2ray-linux-64.zip
                #下载 v2ray 压缩包到当前目录/root/v2ray
root@tc# unzip v2ray-linux-64.zip        #解压缩，然后得到一堆文件
```

这时先不急，先创建 v2ray 要用到的几个路径：

```
root@tc# mkdir  /etc/v2ray/          #配置文件目录
root@tc# mkdir  /usr/bin/v2ray/      #主程序目录
root@tc# mkdir  /var/log/v2ray/      #日志文件目录
```

然后再复制/root/v2ray/里的相关文件到相应位置

```
root@/root/v2ray/# cp v2ray  /usr/bin/v2ray/v2ray        #主程序文件，可执行程序
root@/root/v2ray/# cp v2ctl  /usr/bin/v2ray/v2ctl        #辅程序，可执行程序
root@/root/v2ray/# cp geoip.dat  /usr/bin/v2ray/geoip.dat        #内地 ip
root@/root/v2ray/# cp geosite.dat  /usr/bin/v2ray/geosite.dat    #内地 site
root@/root/v2ray/# cp systemd/v2ray.service  /etc/systemd/system/v2ray.service  #服务文件
```

(4.2)创建配置文件（vi /etc/v2ray/config.json）

v2ray 服务端配置如下：（config.json 文件内容，如需复制，请不要把#注释复制进去）

```
{
  "log": {                    #日志相关配置，日志文件不用事前创建，程序会自己生成
    "access": "/var/log/v2ray/access.log",
    "error": "/var/log/v2ray/error.log",
    "loglevel": "warning"
  },                  #不要漏了逗号
  "inbounds": [{
    "port": 3858,             #v2ray 进程监听 3858 端口，防火墙上不开放此端口，只供本地的 nginx
    "protocol": "vmess",                                        # 进程分流时访问
    "settings": {
      "clients": [            #客户端的验证 Id 相关配置
        {
          "id": "ca9c5d6c-5fee-4e02-929b-818eb077b939",       #这个 uuid 可以更改，相当于密码
          "level": 1,                 #这个 level 不能改，只能用 1
          "alterId": 62               #额外的 id，也相当于密码的一部分，可以改为 1-64 之间的数
        }
      ]
    },
  "streamSettings": {         #入站流量相关配置
    "network": "ws",          #配置为使用 WebSocket 伪装
    "wsSettings": {
          "path": "/vtest"       #分流的路径 /vtest 必须和 nginx 里的 location 一致
          }
  }
  }],
  "outbounds": [{         #出口路由，表示代理服务进行代理工作时走的出口，这里不用改了
    "protocol": "freedom",
    "settings": {}
  }]
}
```

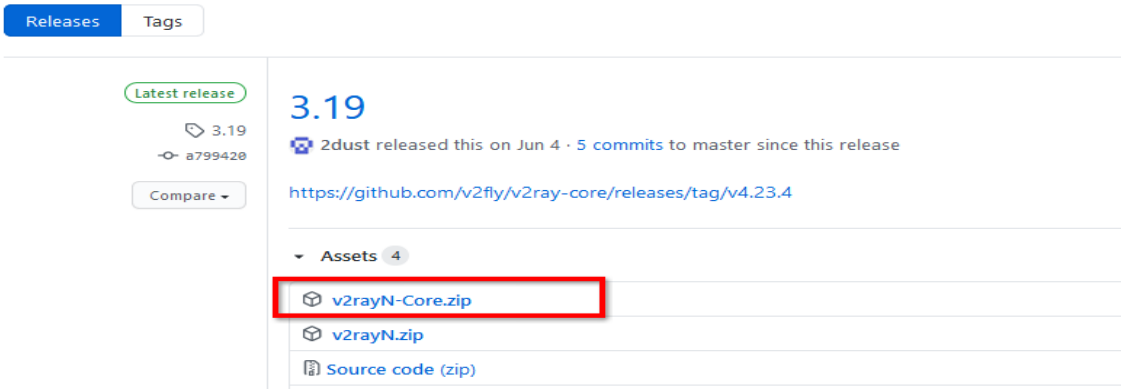
保存后，

```
root@tc# systemctl  start v2ray      #启动服务进程
root@tc# systemctl enable v2ray      #随开机自启
root@tc# systemctl status v2ray      #查看 v2ray 状态
root@tc# ss -ano | grep 3858         #查看 v2ray 是否正确监听端口
```

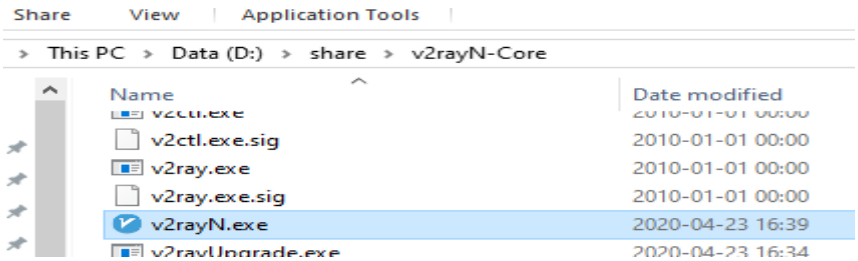
确定 nginx 分流没问题以及 v2ray 服务运行了，就可以配置客户端了。

五、v2ray 客户端的安装及配置（以 windows 为例）

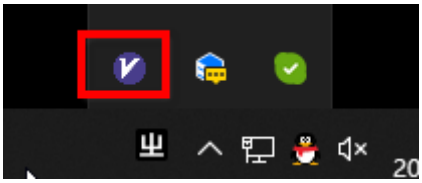
使用 v2rayN 作为 windows 的客户端，它可运行在 windows 7 及之后的任一 windows 系统

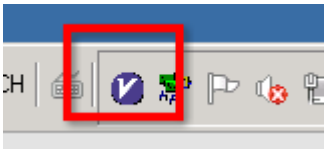


在 https://github.com/2dust/v2rayN/releases 里查看各版本，选择较新版本的 v2rayN-Core.zip 这个文件进行下载，下载后再解压，进入解压目录，看到一个 v2rayN.exe 的可执行文件，运行它就可以



然后在最右下角的系统状态栏托盘里找到 v2rayN 的图标

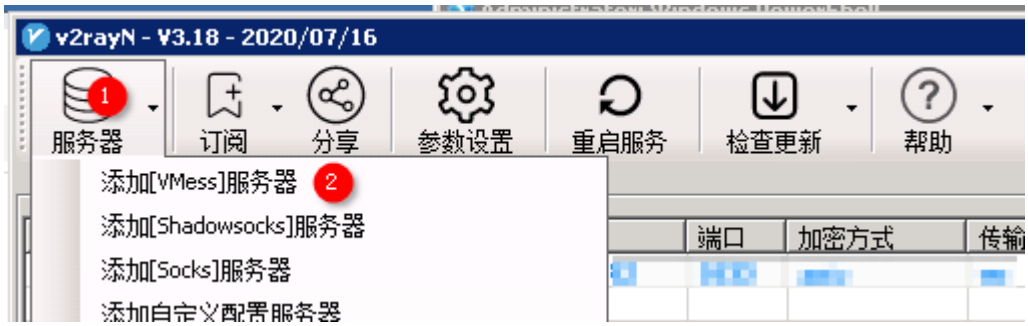


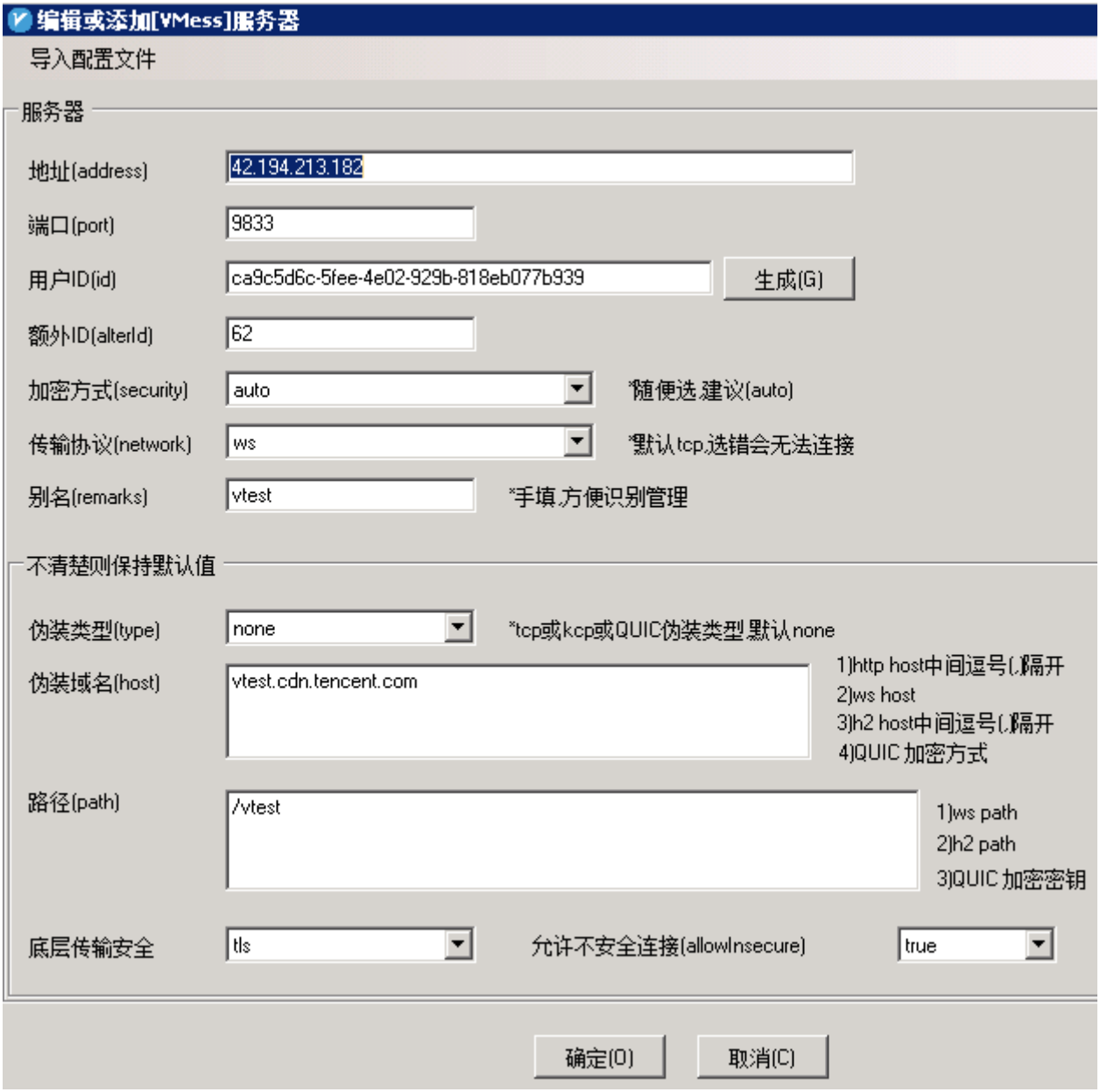


单击它，打开主界面

在主界面左上方工具栏里点击“服务器”→“添加 VMess 服务器”

然后配置如下：



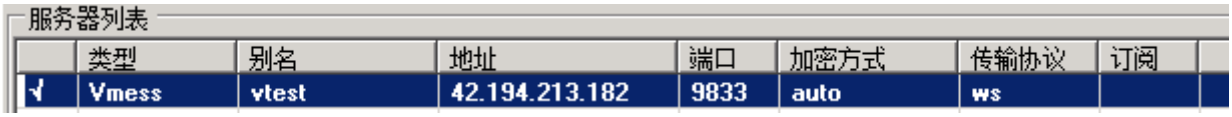


用户 id 那里，要填写服务端里的那个 clients 下的 id，额外 id 为 alterid，path 路径一致，

```
"clients": [          #客户端的验证 Id相关配置
  {
    "id": "ca9c5d6c-5fee-4e02-929b-818eb077b939",
    "level": 1,                #这个 level 不能改，只
    "alterId": 62              #额外的 id，也相当于密
  }
```

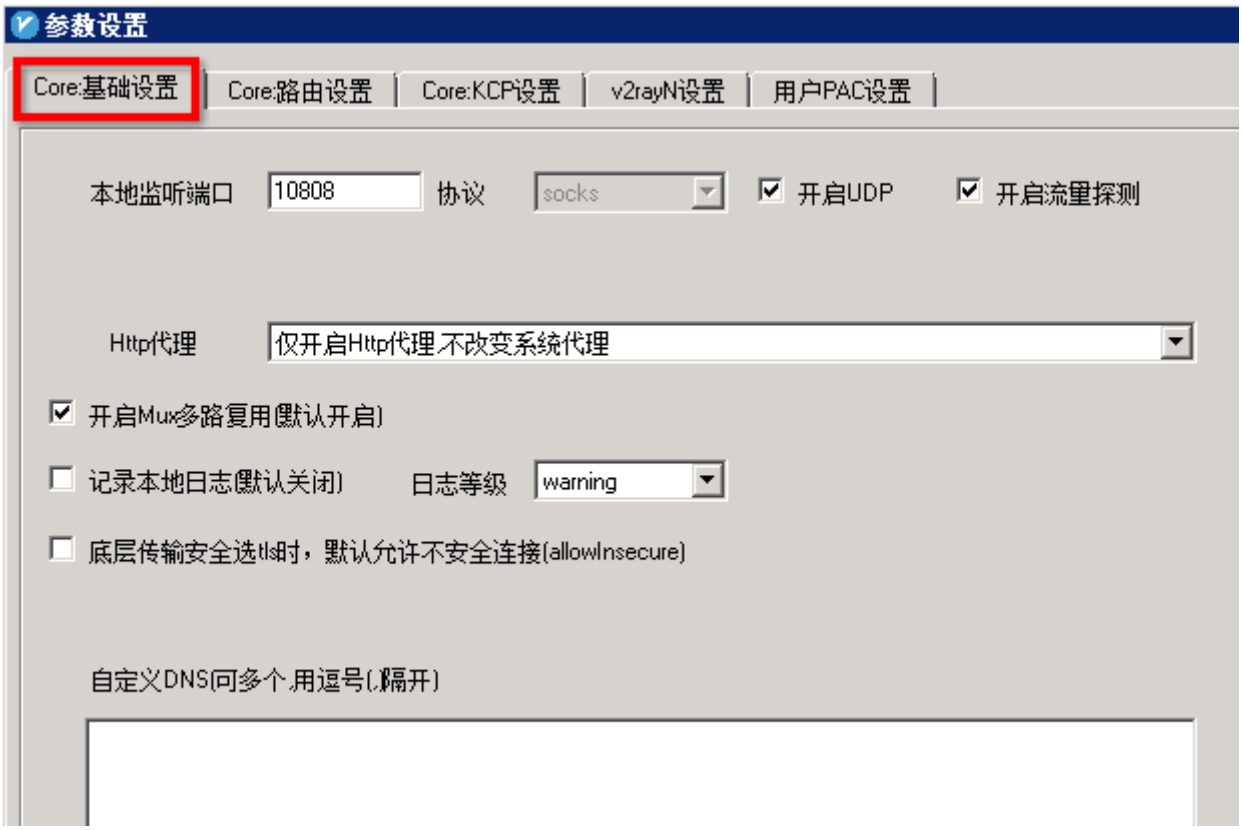
底层传输安全为 tls,允许不安全连接，因为服务端的 ssl 证书是我们自己生成的，不是由受信任的 CA 去签名的，所以要允许不安全的连接，除非我们把服务端的 ssl 证书安装到用户的操作系统里。

添加了服务器后，在主界面里会看到添加成功的服务器，



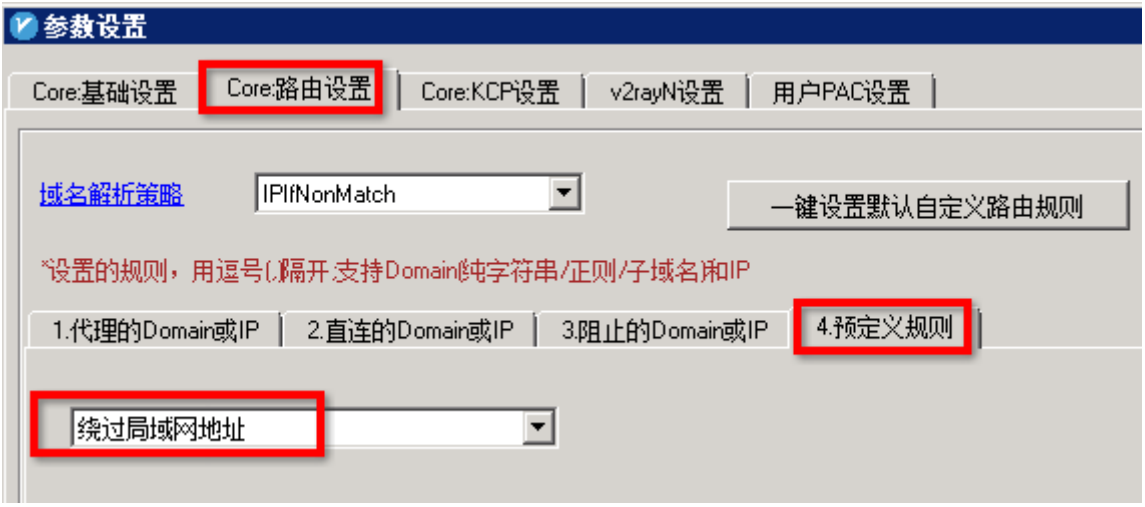
最后再配置一下 v2ray 客户端的代理方式，

点击 v2rayN 主界面工具栏的“参数设置”



仅开启 Http 代理，不改变系统代理，这时它只会监听本地的 10809 端口，要走代理的流量请发往 127.0.0.1:10809 tcp 端口即可。（浏览器等系统代理要另外配置，可以手动写，可以用 pac 脚本

https://static.gtgold.com/ie.pac ）



然后其他的可以不用配了，可以把测试本地上网 ip 的网站放在 pac 文档里，允许它走代理，最后测试一下代理是否成功。（默认时，此 v2rayN 客户端程序的开机自启是要求有用户登录 console 会话的，未作为 services 随 windows 系统启动！）

六、自动配置代理 pac 脚本文件

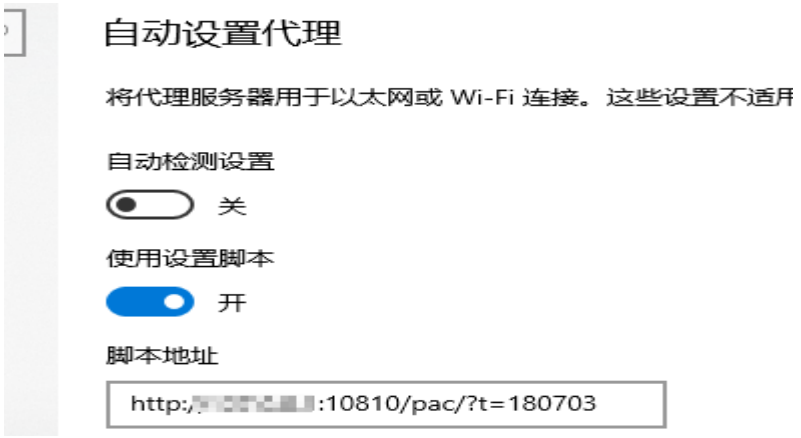
这个 pac 脚本文件就是代理自动配置文件，为文本文件，后缀一般为.pac，内容就是 js 脚本代码，主要实现一个函数：function FindProxyForURL(url, host)
就是根据 url 或 host 决定是否使用代理，以及使用哪个代理，
返回的值有三种：DIRECT, PROXY, SOCKS

DIRECT 表示不使用代理，直接访问

PROXY 表示使用 http/https 代理，后边紧跟代理的 ip 及端口号

SOCKS 表示使用 socks 代理，后边紧跟代理的 ip 及端口号

代理配置脚本文件可以在本地，填写时就写： file:///D:/xxx.pac
也可以位于某个网站上，如： http://xx.com:8888/xxx.pac 一般还可以带个参数，用于验证用户身份
http://xx.com:8888/xx.pac/?t=9734924279



简单的 pac 代理自动配置文件内容示例:

```
function  FindProxyForURL(url, host)
{
   url=url.toLowerCase();        //统一转为小写，方便匹配
 if (shExpMatch(url, "*bank*") ||
   shExpMatch(url, "*xxx*") ) {    //如果访问的 url 中匹配了 bank 或 xxx 字符的
      return  "PROXY  127.0.0.1:10809; DIRECT";  //就返回此信息
 }
 if (shExpMatch(url, "*apple*") ||
   shExpMatch(url, "*yyy*") ) {    //如果访问的 url 中匹配了 apple 或 yyy 字符的
      return  "SOCKS  127.0.0.1:10808; DIRECT";   //就返回此信息
 }
 return  "DIRECT";     //若什么也没匹配的 url，就直接访问，不走代理
}
```

返回结果解析:

PROXY 127.0.0.1:10809; DIRECT 表示走 http/https 代理，代理 ip 为 127.0.0.1 端口号 10809，表示使用本地的 v2rayN 代理客户端监听的 Http/https 代理端口，如果代理无响应或不通则走 direct 直连，就是本机直接访问，不走代理了。

SOCKS 127.0.0.1:10808; DIRECT 表示走 **socks** 代理，代理 ip 为 127.0.0.1 端口号 10808，表示使用本地的 v2rayN 代理客户端监听的 Socks5 代理端口，如果代理无响应或不通则走 direct 直连，就是本机直接访问，不走代理了。

2020-07-17