前两天，ShadowsocksR作者做了一个 公益站，同时演示了 如何将ShadowsocksR服务端的流量伪装成正常网站的流量，有点意思，写一点简单的教程。

如果只是单纯的去伪装ShadowsocksR流量，那么不需要安装Nginx/Apache这些功能多但是配置相对也复杂一点的HTTP服务器，用Caddy就好了。

本文章主要是实现：

> 首先我们要知道 SSR 的混淆插件中的 **HTTP HTTPS** 混淆伪装均为 **仍-伪装**，仅仅是模拟为 HTTP HTTPS 访问流量，容易被主动探测识破，况且很多人连混淆参数都不填，更是进一步降低其伪装混淆安全性，而该教程就是为了**加强混淆伪装，使 仍-伪装 变成 真-伪装**！

访问 http://xxx.xx 域名(http 80端口)，能正常看到网站内容，但如果你链接这个网站服务器上面的ShadowsocksR服务端账号(也是80端口)，也能正常使用，实现了同端口网站和ShadowsocksR共存。

而对于墙或者运营商来说，当他们检测你传输的流量特征的时候，会看到你的流量是去访问海外服务器的 80 端口，就算跟踪过去，也会发现是一个正常的海外网站（随便搞一个静态的 HTML 英文页面），一个没有敏感词的海外英文网站，墙和运营商是不会去注意的。

所以就实现了伪装ShadowsocksR服务端流量的功能，每天国内访问海外网站(80/443端口)的流量非常庞大，在这个伪装中，可以很大程度的降低被发现(匹配流量特征)的几率，所以有兴趣的可以折腾玩玩。

文章里的原理：

你链接你的服务器 **80端口(SSR服务端端口)**，然后就会链接到ShadowsocksR服务端，ShadowsocksR服务端会判断你是**链接SSR，还是访问网站**，如果是访问网站，就会把你的流量数据**重定向到网站监听的端口(2333端口)**，然后你就显示网页了。

> **注意：**如果要用这个同端口伪装功能，请不要开启 **协议(origin)和混淆插件(obfs)的兼容原版模式(_compatible)，不要使用原版SS链接！**
>
> 同时，**Shadowsocks-libev版服务端的simple-obfs功能，并不能实现本教程所说的同端口共存功能！**

推荐一个伪装配置

这个伪装配置不需要你服务器上面搭建网站，而是把HTTPS访问的网页信息转发到微软的网站，实现伪装HTTPS访问微软！

目前已知这种设置方式可以**有效降低 IP 被墙几率（非 100%）**。

```
# 服务端端口设置为：
"server_port": 443,
# 客户端服务器端口填写：443

# 服务端混淆插件设置为：
"obfs": "tls1.2_ticket_auth",
# 客户端混淆插件选择：tls1.2_ticket_auth 或 tls1.2_ticket_fastauth

# 服务端 redirect 设置为：
"redirect": ["*:443#HK2SCH130083c18593.wns.windows.com:443"],
# 客户端混淆参数填写：HK2SCH130083c18593.wns.windows.com
```

至于其他的加密方式、协议插件等都不影响什么都行。

SSR 客户端的混淆参数记得填写为：**HK2SCH130083c18593.wns.windows.com**

前提准备

其实也没什么要准备的，就是你需要有个域名，然后将域名解析为你的SSR服务器IP

> *例如：*
>
> **toyoo.pw** 域名，记录名为 **@** ，解析类型为 **A**，解析IP为 **服务器IP**。
>
> **abc.toyoo.pw** 域名，记录名为 **abc** ，解析类型为 **A**，解析IP为 **服务器IP**。

免费域名申请教程：数你申请.tk/.ml/.cf/.gq/.ga等免费域名

Caddy 配置步骤

安装 Caddy

Caddy 是 Go语言编译好的二进制程序，所以只有一个 Caddy 文件（还需要生成一个配置文件），但是为了管理方便，所以我做了个一键脚本。

```
wget -N --no-check-certificate https://raw.githubusercontent.com/ToyoDAdoubiBackup/doubi/master/caddy_install.sh && chmod +x caddy_install.sh && bash caddy_install.sh install http.filemanager
```

安装Caddy成功后，我们就继续下面的步骤。

配置 Caddy

Caddy的特点之一就是，配置文件非常的简单，继续下面看就知道了。

> 注意：以下示例域名皆为 toyoo.pw 。请注意更换为自己的域名，并做好域名解析。

HTTP（80端口）

假设你要伪装 **80端口**（ShadowsocksR服务端端口为 80），那么你可以在 Caddy中随便配置一个监听端口。

点击展开 查看更多

比如的监听为 2333 端口（可以自己改）。

```
# 以下全部内容是一个整体，是一个命令，全部复制粘贴到SSH软件中并一起执行！
echo 'http://toyoo.pw:2333 {
root /usr/local/caddy/www/ssr
timeouts none
gzip
}' > /usr/local/caddy/Caddyfile
```

HTTPS（443端口）

如果你想要伪装 **443端口**（ShadowsocksR服务端端口为 443），那么你可以在 Caddy中随便配置一个监听端口。

点击展开 查看更多

比如的监听为 2333 端口（可以自己改）。

如果你有 SSL证书和密匙的话，把 **SSL证书(xxx.crt)和密匙(xxx.key)文件**放到 /root 文件夹下（也可以是其他文件夹，自己改下面代码），然后这样做：

```
# 以下全部内容是一个整体，是一个命令，全部复制粘贴到SSH软件中并一起执行！
echo 'https://toyoo.pw:2333 {
root /usr/local/caddy/www/ssr
timeouts none
tls /root/xxx.crt /root/xxx.key
gzip
}' > /usr/local/caddy/Caddyfile
```

如果你没有 SSL证书和密匙，那么你可以这样做：

下面的 **xxxx@xxx.xx** 改成你的邮箱，同时需要注意的是，**申请 SSL证书前，请务必提前解析好域名记录(解析后最好等一会，以全球生效)，否则 Caddy会申请并配置失败！**

```
# 以下全部内容是一个整体，是一个命令，全部复制粘贴到SSH软件中并一起执行！
echo 'https://toyoo.pw:2333 {
root /usr/local/caddy/www/ssr
timeouts none
tls xxxx@xxx.xx
gzip
}' > /usr/local/caddy/Caddyfile
```

如果一切正常，那么Caddy会自动帮你申请 SSL证书并配置好，而且会定时续约SSL证书 和 强制 http重定向至https ！

然后我们新建一个 虚拟主机文件夹

```
mkdir /usr/local/caddy/www && mkdir /usr/local/caddy/www/ssr
```

然后去网上随便下载一个 **HTML的网页模板**（很好找的，这类模板非常多，随便找个英文模板就好了），解压后把网页文件通过SFTP 上传到刚才新建的虚拟主机文件夹中：/usr/local/caddy/www/ssr

然后我们重启 Caddy，当然这个时候我们还只能通过 http://toyoo.pw:2333 来访问我们的网站，我们还需要继续配置ShadowsocksR服务端。

```
/etc/init.d/caddy restart
```

使用说明

启动：/etc/init.d/caddy start

停止：/etc/init.d/caddy stop

重启：/etc/init.d/caddy restart

查看状态：/etc/init.d/caddy status

查看Caddy启动日志： tail -f /tmp/caddy.log

Caddy配置文件位置：/usr/local/caddy/Caddyfile

ShadowsocksR 配置步骤

打开你的配置文件（假设你的ShadowsocksR服务端安装在 /root 文件夹内）：

```
nano /root/shadowsocksr/user-config.json
```

然后我们会看到配置文件，我们需要找到 server_port 和 redirect 参数 并修改就行了

修改完成后，按 **Ctrl+X** 退出，会提示你是否保存，输入 **y** 代表保存，然后会提示你要保存的文件名，直接回车覆盖。输入 **n** 代表不保存，直接退出。

HTTP（80端口）

按照上面Caddy的HTTP配置（监听2333端口），

```
# 把 server_port 参数改成这样：
"server_port": 80,

# 把 redirect 参数改成这样：
"redirect": ["*:80#127.0.0.1:2333"],
```

HTTPS（443端口）

按照上面Caddy的HTTPS配置（监听6666端口），

```
# 把 server_port 参数改成这样：
"server_port": 443,

# 把 redirect 参数改成这样：
# 如果是配置的HTTP转HTTPS，那么也是使用下面这个配置（只需要设置一个 443的转发即可）
"redirect": ["*:443#127.0.0.1:2333"],
```

多端口配置

如果你的Caddy和SSR服务端同时配置了多个端口（比如 888和666），

```
# 那么把 redirect 参数改成这样：
"redirect": ["*:888#127.0.0.1:2333", "*:666#127.0.0.1:6666"],
```

然后重启ShadowsocksR服务端（假设你的ShadowsocksR服务端安装在 /root 文件夹内，并且是单用户）：

```
bash /root/shadowsocksr/shadowsocks/run.sh
```

> **注意：**如果要用这个同端口伪装功能，请不要开启 **协议(origin)和混淆插件(obfs)的兼容原版模式(_compatible)，不要使用原版SS链接！**
>
> 同时，**Shadowsocks-libev版服务端的simple-obfs功能，并不能实现本教程所说的同端口共存功能！**

然后现在你可以尝试访问 http://toyoo.pw ，会发现不需要加端口号（ http://toyoo.pw:2333 ）也可以正常访问了！

客户端混淆设置

这时候为了加强该伪装的作用，还需要客户端**混淆插件和混淆参数**设置一番。

如果你是伪装的 80 端口，请选择 http_simple 混淆插件，混淆参数填写你的域名(示例 toyoo.pw)。

如果你是伪装的 443 端口，请选择 tls1.2_ticket_auth 或 tls1.2_ticket_fastauth 混淆插件，混淆参数填写你的域名(示例 toyoo.pw)。

SSR客户端的服务器地址填写 你的域名(示例 toyoo.pw)

SSR客户端的服务器端口填写 80 或 443 端口(取决于服务端端口)。

所以这个就实现了，伪装 ShadowsocksR 服务端流量为 正常海外网站流量的目的，同时也可以说实现了 **网站和ShadowsocksR 同端口共存**。

其他说明

启动显示成功，但是实际未运行

点击展开 查看更多

因为 服务脚本判断的问题，只判断了nohub是否运行 Caddy成功，但没有判断 Caddy 是否保持正常运行。

你可以理解为，nohub成功启动了 Caddy，但是 Caddy因为配置文件错误等原因，启动后又退出了。

所以这种情况下，你应该去查看启动日志：

```
tail -f /tmp/caddy.log
```

Nginx配置说明

点击展开 查看更多

至于 Nginx的话，你只需要找到你的 虚拟主机配置文件（lnmp.org 一键包的位置在/usr/local/nginx/conf/vhost/域名.conf ），然后把监听端口改成像上面 Caddy配置里监听的端口一样。

```
# Nginx 默认监听 80 443。
listen 80;
listen 443;
```

端口 **80或者443** 可以改成其他的，比如 **2333和6666**，然后ShadowsocksR配置文件和上面的步骤还是一样。

参考资料：https://breakwa11.blogspot.ru/2017/01/shadowsocksr-mu.html?m=1
