SS详解（一）：原理

很多人根本搞不清楚SS几种模式是什么东西，基本上按照自己那种能勉强弄通就用哪种，稍微遇到一点问题动不动就怪固件不稳定，说句老实话，每次遇到这种情况，我心里就是千万匹草泥马奔腾而过。

好吧，之前不懂不怪你，是因为几乎没有几篇内容是扫盲的，大部分都是针对专业人员的讨论，下面打算从最粗浅的内容开始介绍，专业人员请略过此篇文章内容，因为对你毫无帮助。

首先，我们看看SS到底是什么东西，SS全名ShadowSocks，是一位在知乎网站工作的大神开发的一套针对DPI（深度包检测）系统弱点开发的一套网络加密工具。它通过抹除网络数据通讯的特征，来让检测系统无从下手，从理论上来说，SS的处理模型并非绝对没有弱点，但是，这个弱点要利用则难度非常大，用个比喻来说，如果你在鱼缸里面，要找到一条特定颜色的小鱼，会很简单，但是，当面对整个海洋来说，要找到这条鱼几乎是不可能的。SS就是利用这样一个原理，把自己的数据通讯模式变得毫无特征，所以，当拦截系统需要对SS进行拦截的时候，需要对每一个水滴进行检测，而目前任何一个国家的拦截系统的运算能力，都几乎是不可能做到的。

我们稍微了解下，SS到底是如何做到这种功能的，说穿了道理也很简单，传统的VPN也好，网络加密也好，首要考虑的是安全原则，也就是：即便你知道了我在传输数据，你也一样无法解开我传输的内容，所以，大家根本不太在乎到底别人是否能够发现自己在传输加密的数据，因为通过RSA的天才算法，使得在传输过程中的数据解密几乎在目前的状态下是不太可能的，甚至在整个加密传输开始前，大家甚至可以大喇喇地使用明文来传输一些打招呼的信息，用来告诉对方，接下来我准备用闽南话和你通话了，拦截系统则刚好利用了这点，它们并不在乎听懂你说什么，但是，只要你一说，接下来用闽南话开始通话这几个字，就开始搞鬼，或者直接把你的通话线路掐掉，或者在你前面说的话前面插几句四川话，让听的人觉得：说什么鬼啊，明明是四川话，说好的闽南话呢？肯定是骗子，不理他，于是，你的加密传输就中断。所以，在几轮的系统升级后，到了2016年春季，几乎所有的VPN协议以及网络加密传输协议，均可用不同的方式被破坏，导致整个通讯出现中断。

但SS呢？它利用了一点：也就是通话双方都是可预先约定的，例如约定双方讲的就是闽南话，你要听懂我说的，就要找个懂闽南话的人和我聊，这样，从第一句开始说的话开始，就是用闽南话开始说，而听的呢，则一早准备好一个懂闽南话的来交流，这样通讯双方可以互相听懂，而偷听的人由于并不知道双方约定是用什么语言的，于是就傻眼了，而在网络传输中可用的语言几乎是无限种类的，所以就造就了SS神奇的能力，于是偷听的人心想：你妹的，看老子怎么治你！最终，SS的作者会被拉去喝茶被迫删代码了。可惜机关算尽，github的源代码的复制能力，不是作者删了自己的代码就可以阻止一套代码传播的。

つづく

# SS详解（二）：SS 性能

SS目前有很多不同语言的版本，最原始的来源是python 的，也有 golang 的，通常我们在路由中使用的是 C 语言版本的。那为什么路由中要用C语言版本的SS？

我们首先来看看智能路由到底是什么？目前大部分的所谓智能路由，其实就是一台低配的电脑，但这个低配到底有多低呢？拿我们这个固件支持最多的机型 MT7620来举例，它的运算能力为BogoMIPS：382，而最古老的i3 530，CPU 的运算能力为 5850，Pentium G3220 3.00GHz ，运算能力为 5986.0 BogoMIPS，也就是路由的CPU大约只有普通CPU的1/15 的处理性能。拿加密中最常见的AES加密模式，普通的i3 CPU大约每秒可以进行70Mbits左右的数据，而MT7620，每秒就只剩下5Mbits了。所以，在路由上，大部分的高级语言程序跑起来都会非常吃力，尤其是脚本型的语言，例如 python和php，而越接近底层的低级语言，例如C，运行起来就还算凑合。

所以，当你选择SS加密模式的时候，要在路由上获得较快的速度，第一点要素，就是需要选取一款能在路由上跑得顺畅的加密模式，例如由google提供的 chacha20 系列，基本上它能在MT7620 上跑出40Mbits/s左右的速度。而加了更多运算的同类型通道，例如二次元宅男们喜欢的 SSR，由于加了额外的处理运算，使得它的性能大打折扣，最终只能在PC机上跑跑而已。

同理，很多应用你想移植到路由上跑，首先，要考虑下内存大小的因素，然后，就是CPU能力，很多你在PC上、手机上能跑的应用，搬到路由上，则会有很多问题。

つづく

# SS详解（三）：SS模式：gfwlist

SS 使用的方法有很多结构和模式。那么，我先从我自己个人最喜欢的模式以及这个固件的功能来介绍。

1）ss-redir + gfwlist

个人认为，这种模式是最稳定、最可靠、最少干扰正常网络使用，对于路由器的性能影响最小。它通过一个名为 gfwlist 的列表，指出了那些需要从SS通道走的域名。

路由中内置了一个常用的被墙域名列表，这个gfwlist 列表位于固件目录中的 /etc\_ro/basedomain.txt ，系统启动后，会在创建一个 /etc/storage/basedomain.txt 的链接指向 /etc\_ro/basedomain.txt 。如果你想建立一个专用的常用域名列表，直接删除这个文件，重新建立就好了。

SS在启动时候，脚本会自动从这个列表结合 /etc/storage/shadowsocks\_mydomain\_script.sh 这个文件的域名内容（这个文件就是我们在路由界面添加的自定义域名列表），生成一个适用于 dnsmasq 的配置文件，存放于 /tmp/ss/dnsmasq.d 中，文件名为 r.gfwlist.conf 。

当 r.gfwlist.conf 自动生成完毕之后，dnsmasq 会重启激活这个配置，并且启动 pdnsd 以及 ss-redir 。

让我们看看在这种模式下，我们访问 google.com 的整个流程是怎么样的。

1. 我们在PC浏览器输入 google.com ，浏览器查询域名的IP
2. 域名IP查询会优先从pc的 hosts 文件查询，所以，要确保你要访问的域名没有被你的hosts文件解析出来，这个是少数人遇到gfwlist 模式无法正常工作的常见原因之一。
3. 你的电脑会从你网络设定的DNS服务器查询域名的IP，如果你电脑上的网络设置 dns server不是指向这个路由，那么，整个功能是无法使用的，这也是大多数人遇到 gfwlist 无法工作的最常见原因之一。
4. 当你的电脑向路由查询 google.com IP 的时候，路由中的 dnsmasq 接收到这个查询指令，然后发现 google.com 在 r.gfwlist.conf 的列表中，并且列表指出这个域名需要通过 pdnsd 的端口查询，因此，dnsmasq会把域名查询转给 pdnsd ，让它来做解析。
5. 我们的固件中，采用的是pdnsd 来把 udp 的域名请求转换为 tcp域名请求的，目前国内的网络环境，udp的域名请求被大量投毒，而tcp的几乎没有，所以，我们在pdnsd 设置了 opendns 和 google 的dns作为域名解析服务器，为了获得更加合理的域名解析，我们需要在固件UI里面设置 中设置 pdnsd 通过 SS代理访问，这样dns 服务器才能返回离你ss-server 最近的IP，而不是离你的PC 最近的IP，当然，你硬是要觉得pdnsd 直连才让你舒服，我也没有办法。
6. 当 pdnsd 获得了 google.com 的解析后，会转交给dnsmasq ，dnsmasq 再次检查 r.gfwlist.conf，发现 google.com 还有一个ipset 的设置，于是，它会把这个域名的ip 放入到 ipset 的 gfwlist 列表中，然后把ip返回给你的电脑，以上1~6的步骤看起来好复杂，但整个处理时间大约为0.2秒左右。
7. 当你的电脑获得了 google.com 的IP后，开始向你的电脑网络设置中的网关发出向google ip 的连接请求，如果你的网关设置不是指向这个路由，那么，这也是少数人遇到 gfwlist 不能工作的情况。
8. 当路由收到有发向google IP 的网络连接请求后，开始核对 ipset gfwlist 的ip列表，如果有符合的，直接重定向到 ss-redir 的透明代理端口。
9. ss-redir 把数据进行加密，发给远端的 ss-server。
10. ss-server解密数据包，发出连接请求，获得服务器的数据响应，加密，返回给路由的 ss-redir。
11. ss-redir 解密，把数据返回给你的电脑，这样，整个连接就完成了。

基本上整个gfwlist 模式运作的，可以遇到的问题，就是上面红字标明的问题。那么，我们怎么来检查电脑上的设置是否可以正常工作呢？

1. nslookup  
   这条指令可以查询域名的解释过程是否正常。例如：
2. nslookup www.facebook.com
3. Server: 192.168.199.1
4. Address: 192.168.199.1#53
5. Non-authoritative answer:
6. www.facebook.com canonical name = star-mini.c10r.facebook.com.
7. Name: star-mini.c10r.facebook.com

Address: 31.13.95.36

首先，这里我们可以检查 Server 是否你的路由器IP，如果不是，就自己检查哪里的设置错了。  
然后，我们可以看看 www.facebook.com  里面是否包含了 star-mini 这个 cname 的内容，这个是最容易判断域名解释是否被投毒的情况。被投毒的域名是不会出现这个内容的。

1. ipconfig   
   用这条指令检查你的IP设置中，网关是否指向路由的IP，如果不是，则自己检查和改回来。
2. 用telnet 来检查你的ss-server 是否可以连接得上。

好了，大概就这几招了。按照这种方法，几乎没有见过 gfwlist 翻墙不顺利的，除非你连ss-server 的端口、密码、加密方式这些都填错。

つづく