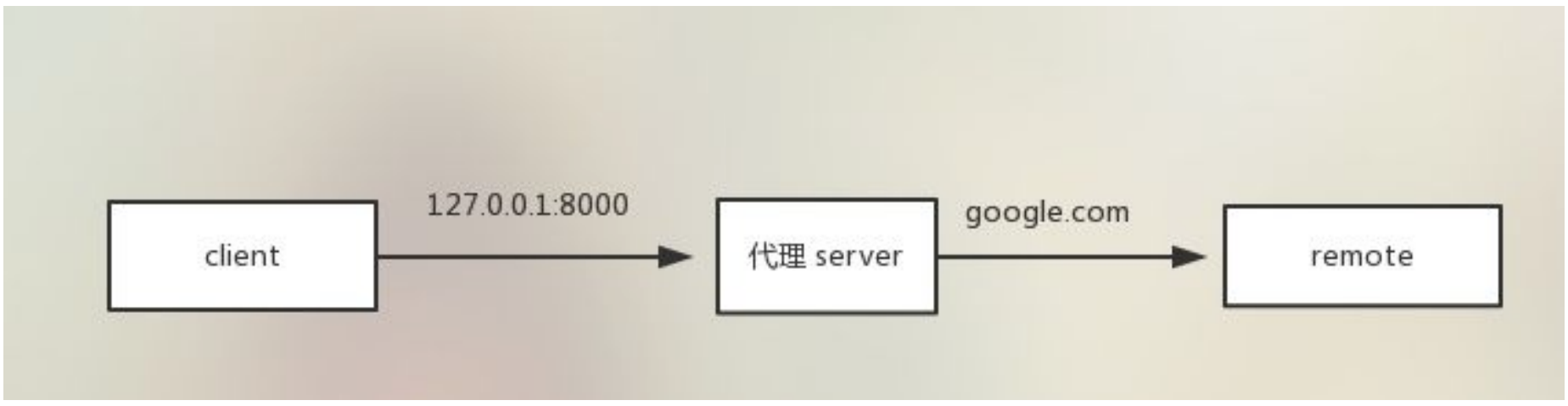
由浅入深写代理(3) -socks5 代理

facert
微信公众号「程序化思维」
+ 关注他

本文讲的是如何写一个 socks5 代理，其实 shadowsocks 的代理也是 socks5 协议的，所以 socks5 代理也是本系列教程的一个重点。

首先放出 socks5 协议的 [rfc](ietf.org/rfc/rfc1928.tx...)，socks5 协议很简单，SOCKS5 协议并不负责代理服务器的数据传输环节，此协议只是在 C/S 两端真实交互之间，建立起一条从客户端到代理服务器的授信连接。

0x01 sock5 代理结构图



0x02 socks5 协议分析

认证阶段

首先客户端需要和服务端有个握手认证的过程，可以采用 *用户名/密码* 认证或者无需认证方式。

格式如下 （*数字表示位数*）

```
+----+----------+----------+
|VER | NMETHODS | METHODS  |
+----+----------+----------+
| 1  |    1     |  1~255   |
+----+----------+----------+
```

- VER 字段是当前协议的版本号，也就是 5；
- NMETHODS 字段是 METHODS 字段占用的字节数；
- METHODS 字段的每一个字节表示一种认证方式，表示客户端支持的全部认证方式。

```
0x00: NO AUTHENTICATION REQUIRED
0x01: GSSAPI
0x02: USERNAME/PASSWORD
0x03: to X’7F’ IANA ASSIGNED
0x80: to X’FE’ RESERVED FOR PRIVATE METHODS
0xFF: NO ACCEPTABLE METHODS
```

服务端返回格式

```
    +----+--------+
    |VER | METHOD |
    +----+--------+
    | 1  |   1    |
    +----+--------+
```

一般情况下服务端返回两种情况

0x05 0x00：告诉客户端采用无认证的方式建立连接；

0x05 0xff：客户端的任意一种认证方式服务器都不支持。

举个例子， 服务器无需认证的情况如下

```
    client -> server: 0x05 0x01 0x00
    server -> client: 0x05 0x00
```

连接阶段

认证完成，客户端向服务端发送请求：

```
    +----+-----+-------+------+----------+----------+
    |VER | CMD |  RSV  | ATYP | DST.ADDR | DST.PORT |
    +----+-----+-------+------+----------+----------+
    | 1  |  1  |   1   |  1   | Variable |    2     |
    +----+-----+-------+------+----------+----------+
```

- CMD 字段 command 的缩写：

```
    * 0x01: CONNECT 建立 TCP 连接
    * 0x02: BIND 上报反向连接地址
    * 0x03: 关联 UDP 请求
```

- RSV 字段：保留字段，值为 0x00
- ATYP 字段：address type 的缩写，取值为：

```
    * 0x01: IPv4
    * 0x03: 域名
    * 0x04: IPv6
```

- DST.ADDR 字段：destination address 的缩写， 取值随 ATYP 变化：

```
    * ATYP == 0x01: 4 个字节的 IPv4 地址
    * ATYP == 0x03: 1 个字节表示域名长度，紧随其后的是对应的域名
    * ATYP == 0x04: 16 个字节的 IPv6 地址
    * DST.PORT 字段: 目的服务器的端口
```

服务端返回格式

```
+----+-----+-------+------+----------+----------+
|VER | REP |  RSV  | ATYP | BND.ADDR | BND.PORT |
+----+-----+-------+------+----------+----------+
| 1  |  1  |   1   |  1   | Variable |    2     |
+----+-----+-------+------+----------+----------+
```

- REP 字段

```
* X'00' succeeded
* X'01' general SOCKS server failure
* X'02' connection not allowed by ruleset
* X'03' Network unreachable
* X'04' Host unreachable
* X'05' Connection refused
* X'06' TTL expired
* X'07' Command not supported
* X'08' Address type not supported
* X'09' to X'FF' unassigned
```

举个例子，客户端通过 127.0.0.1:8000 的代理发送请求

```
# request:         VER  CMD  RSV  ATYP DST.ADDR             DST.PORT
client -> server: 0x05 0x01 0x00 0x01 0x7f 0x00 0x00 0x01 0x1f 0x40
# response:        VER  REP  RSV  ATYP BND.ADDR             BND.PORT
server -> client: 0x05 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x10 0x10
```

传输阶段

接下来就开始传输数据，socks5 服务器只做单纯的转发功能

整个过程如下

```
# 认证阶段
client -> server: 0x05 0x01 0x00
server -> client: 0x05 0x00
# 连接阶段
client -> server: 0x05 0x01 0x00 0x03 0x0a b'google.com'  0x00 0x50
server -> client: 0x05 0x00 0x00 0x01 0x00 0x00 0x00 0x00 0x10 0x10
# 传输阶段
client -> server -> remote
remote -> server -> client
...
```

下篇教程用代码实现下 socks5 代理

参考链接：

* Shadowsocks 源码分析--协议与结构

* ietf.org/rfc/rfc1928.tx...

* SOCKS5 协议解析 - 莫邪

发布于 2017-08-20

真诚赞赏，手留余香

赞赏

还没有人赞赏，快来当第一个赞赏的人吧！

计算机网络 socks代理 Python

文章被以下专栏收录

程序化思维

关注专栏

4 条评论

切换为时间排序

写下你的评论...

赞同 49

4 条评论

分享

喜欢

收藏

...