# Squid tcp正向代理功能

**对比其它的代理软件:**

**Squid**

高性能的代理缓存服务器，支持FTP、gopher、HTTPS和HTTP协议。和一般的代理缓存软件不同，Squid用一个单独的非模块化的I/O驱动的进程来处理所有的客户端请求。

在实现正向代理功能时，内网服务器上必须要加上以下参数，才能上外网

http\_proxy = proxy ip:port

https\_proxy = proxy ip:port

**Nginx**

配置简单，功能强大，社区活跃，但不支持正向tcp代理，使用正向七层代理时，如果需要转发https时，需要用到以下3种方法：

1. 和squid一样，添加http\_proxy参数，配置文件同squid
2. 访问https的连接时，不要加https，在域名后面加上:443端口，指定使用443端口转发
3. 代理服务器配置ssl自签名证书，但访问https的连接时需要加上 --insecure 参数

**Gost**

无需安装，使用go语言写的，性能消耗极小，在转发https的连接时遇到同nginx相同的问题，解决方法一致。

**Tingproxy**

专业的老牌正向代理软件，在转发https的连接上遇到同上面一样的问题，解决方法一致

**HAProxy**

不支持正向代理，但可以使用反向代理实现代理上网功能，但需要给每个外网域名配置一个端口，这对内网服务器访问极不方便

**使用squid正向代理功能，实现内网服务器上外网**

**测试环境**

**服务器OS version : CentOS\_x64 7.5**

**Squid 版本： 4.4**

|  |  |
| --- | --- |
| IP | 角色 |
| 10.1.1.21 | 内网服务器1 |
| 10.1.1.22 | 内网服务器2 |
| 10.1.1.50 | 代理上网服务器 |

1. **安装squid**

Squid 4.4 Latest Stable源码官网下载:

<http://www.squid-cache.org/Versions/v4/squid-4.4.tar.gz>

编译安装：

# yum install -y perl gcc autoconf automake make sudo wget libxml2-devel libcap-devel libtool-ltdl-devel

# tar zxvf squid-4.4.tar.gz -C /usr/src/

# cd /usr/src/

# ./configure --prefix=/usr/local/squid4.4/ --includedir=/usr/include --datadir=/usr/local/squid4.4 --libexecdir=/usr/lib/squid --localstatedir=/var/ --sysconfdir=/usr/local/squid4.4

# make

# make install

1. **配置Squid**

这里的配置主要完成三个功能：

1. 正向代理上网功能，包含http和https
2. 内网服务器的黑白名单
3. 外网访问域名的黑白名单

配置文件参数如下：

# egrep -v "#|^$" ./squid.conf

acl Safe\_ports port 80 # http --允许通过代理的端口

acl Safe\_ports port 443 # https

acl CONNECT method CONNECT

acl inside src "/usr/local/squid4.4/allow\_src\_ip" --内网服务器白名单

acl external dst "/usr/local/squid4.4/allow\_dst\_domain" --访问外网域名&ip的白名单

http\_access deny !Safe\_ports --拒绝非80或443以外的端口通过代理

http\_access deny manager

http\_access allow inside --允许内网白名单通过代理

http\_access deny external --允许访问外网的域名

http\_access deny all --拒绝所有域名和内网服务器

http\_port 80 --监听端口

dns\_nameservers 8.8.8.8 --域名解释dns

coredump\_dir /var/spool/squid --访问外网域名加速缓存目录

refresh\_pattern ^ftp: 1440 20% 10080

refresh\_pattern ^gopher: 1440 0% 1440

refresh\_pattern -i (/cgi-bin/|\?) 0 0% 0

refresh\_pattern . 0 20% 4320

# cat allow\_src\_ip --内网白名单

10.1.1.21

# cat allow\_dst\_domain --外网域名白名单

[www.baidu.com](http://www.baidu.com)

api.mch.weixin.qq.com

1. **测试**

上一步配置了内网服务器10.1.1.21可以通过代理服务器访问外网，且配置了只能访问www.baidu.com和api.mch.weixin.qq.com

**注：内网通过代理服务器上外网时，需要在本地添加两个参数**：

export http\_proxy=http://10.1.1.50:80

export https\_proxy=http://10.1.1.50:80

**该两个参数会影响到yum的使用，为了更好的控制上网行为且不影响yum仓库，建议该参数添加至需要连接外网应用的系统帐号上的环境变量文件上($home/.bashrc)。**

**如：现在在部分应用都跑在svcaccttomcat系统用户上，那该参数添加至该用户家目录下的 .bashrc 文件里，此时只有该用户才可以上外网，不影响到其它用户。**

测试步骤：

1、测试内网白名单是否生效

a、在10.1.1.21和10.1.1.22上访问 www.baidu.com和api.mch.weixin.qq.com

B、查看代理服务器日志

2、测试外网域名白名单是否生效

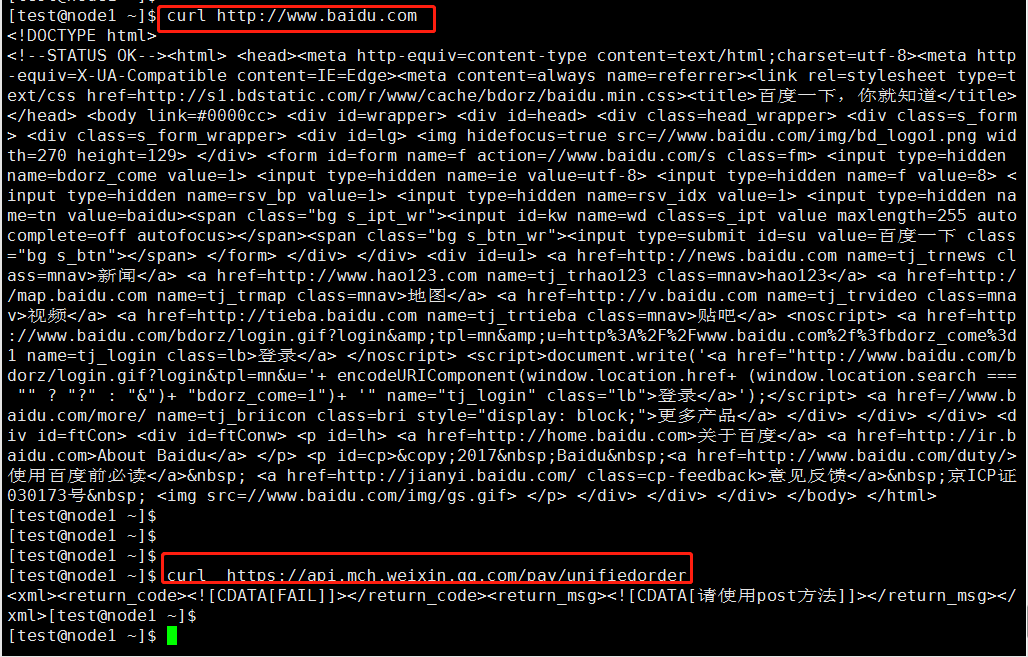
a、分别在以上2台内网服务器访问cn.bing.com

b、查看代理服务器日志

**第一步：测试内网白名单是否生效**

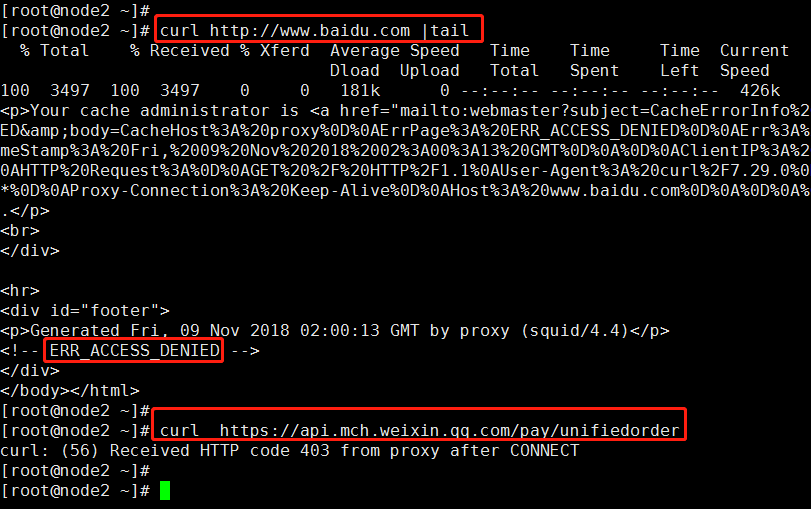
**10.1.1.21服务器**

可以看到访问两个外网网站都返回了数据。

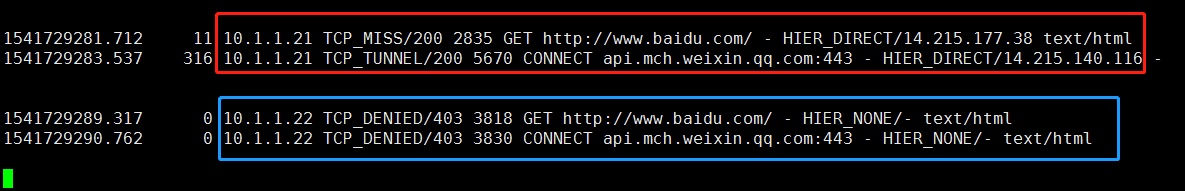


**10.1.1.22服务器**

**可以看到访问百度时被拒绝了，访问api时报403错误**



**来到代理服务器查看日志，发生了什么**



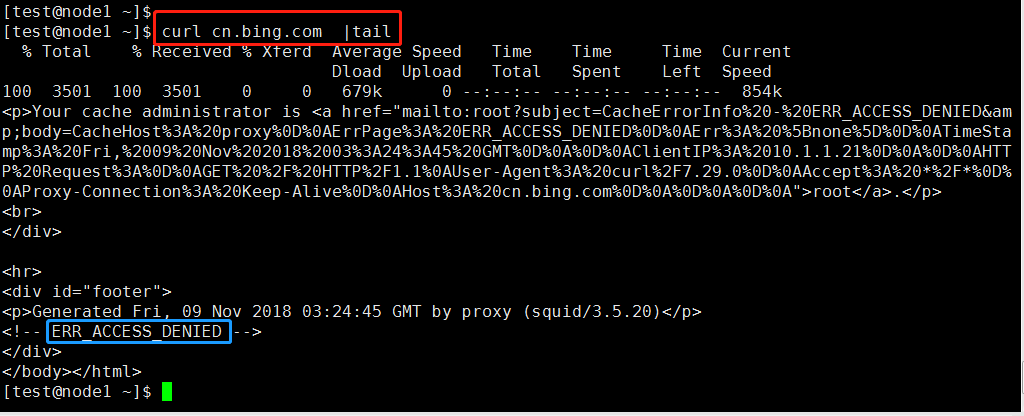
这里可以看到10.1.1.21访问的百度和api都通过了代理

10.1.1.22都显示 TCP\_DENIED 被代理拒绝了

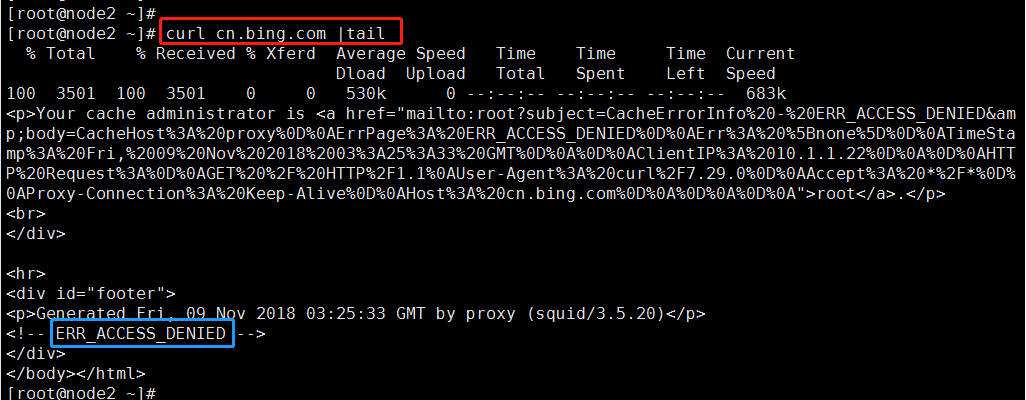
**配置文件的内网白名单已生效**

**第二步：测试外网域名白名单是否生效**

**10.1.1.21服务器**



**10.1.1.22服务器**



**可以看到cn.bing.com 都被拒绝访问了，是因为该域未被加入到白名单里**

另外配置里参数还针对域名解释出来的IP访问也进行了控制。

如：内网服务器使用域名解释出来的IP地址访问外网时，Squid会检测IP属于哪个域名，当该域名未在白名单时，Squid一样的会拒绝放行**。**

1. **结束。**

**-----Kevin**