Squid代理服务器

1 Squid基础服务

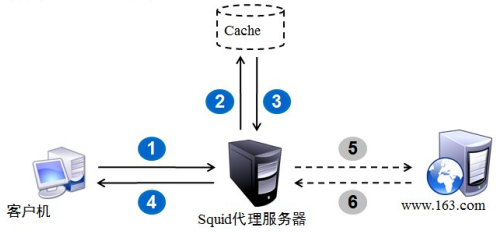
在RHEL5系统中，默认已经安装了squid-2.6.STABLE21-6.e15软件包(如果没有安装则通过RHEL5安装盘中的rpm包安装即可)。本节将介绍缓存代理的工作机制、类型以及Squid服务的主配置文件。

1.1 缓存代理概述

作为应用层的代理服务软件，Squid主要提供缓存加速、应用层过滤控制的功能。

1.代理的工作机制

当客户机通过代理来请求Web页面时，指定的代理服务器会先检查自己的缓存，如果缓存中已经有客户机需要的页面，则直接将缓存中的页面内容反馈给客户机；如果缓存中没有客户机要访问的页面，则由代理服务器向Internet中发送访问请求，当获得返回的Web页面以后，将网页数据保存到缓存中并发送给客户机，如图所示。



http 代理的缓存加速对象主要是文字、图像等静态 web元素， 使用缓存机制后若客户机在不同的时候访问同Web元素，或者不同的客户机访问相同的Web元素时可以直接找代理服务器的缓存中获得结果。这样就大大减少了向Internet提交重复的Web请求的过程，提高了客户机的Web访问响应速度。

由于客中机的Web访问请求实际上是由代理服务器来代替完成的从而可以隐藏用户的真实IP地址，起到一定的保护作用。另一方面代理服务器担任着类似"经纪人"的角色，所以有机会针对要访问的目标，客户机的地址，访问的时间段等进行过滤控制。

2. 代理的基本类型

根据代理服务器的实现方式不同，包括传统代理，透明代理两种常见的代理服务。

➢ 传统代理：也就是普通的代理服务器，在客户机的浏览器，OQ聊天工具，下载软件等程序中必须手动设置代理服务器的地址和端口，然后才能使用代理来访问网络。对于网页浏览器来说，访问网站时的域名解析请求也会发给指定的代理服务器。

➢ 透明代理：提供和传统代理相同的功能和服务，其区别在于客户机不需要指定代理服务器的地址和端口，而是通过默认路由，防火墙策略将Web访问重定向，实际仍然交给代理服务器来处理。重定向的过程对客户机来说是"透明"的，用户甚至并不知道自己在使用代理服务，所以称为"透明代理"。使用透明代理时，网页浏览器访问网站时的域名解析请求将优先发给DNS服务器。

实际应用中传统代理多见于Internet环境，例如为QQ程序使用代理可以隐藏本机真实IP地址，为下载工具使用多个代理可以规避服务器的并发连接限制。而透明代理多见于局域网环境，例如在Linux网关中启用透明代理后局域网主机无需额外设置就可以享受更好的上网速度。

1.2 Squid的配置文件

Squid服务的配置文件位于/etc/squid/squid.conf，其中包含大量的注释内容为相关的配置项提供了详尽的解释和说明。充分了解这些配置行的作用，将有助于管理员根据实际情况灵活配置代理服务。

下面分别讲解其中最常用的几个配置项。

➢ http_port 3128：主要用来指定代理服务监听的地址和端口(默认的端口号为TCP 3128)。如果服务器有多个网络接口，但只希望在其中一个IP地址上提供服务还可以同时指定IP地址，例如"http_port 192.168.1.1:3128"。

➢ cache_mem 64M：指定缓存功能所使用的内存空间大小，便于保持访问较频繁的Web对象。大小单位可使用MB，容量应为4的倍数，建议设置为实际内存的1/4到1/3，具体根据服务器的性能和负载而定。

➢ maximum_object_size 4096KB：允许保存到缓存空间的最大对象(文件)大小，一般以KB为单位，超过大小限制的文件将不会被缓存，而是直接转发给用户。默认的4096KB限制可以满足绝大部分的HTML页面、图片、Flash等Web对象，如果希望代理服务器缓存音频，视频等较大的文件，则应该适当增加此参数的值。

➢ reply_body_max_size 10240000 allow all：允许用户下载的最大文件大小，以字节(byte)为单位。默认设置为0字节，表示不进行限制。其中，all为默认的访问控制列表名，针对任意地址的代理用户。

➢ cache_dir ufs /var/spool/squid 100 16 256：指定缓存数据所使用的目录、容量、子目录个数等相关参数。其中ufs(UNIX File System，UNIX文件系统)是Squid最早使用的缓存文件的格式，也是Squid内建的存储格式类型;/var/spool/squid是缓存数据的默认存放目录；后面三个数字依次表示为缓存缓存目录分配的磁盘空间大小（单位为KB）、一级子目录个数，二级子目录个数。当代理的用户数量较多时可以适当增大缓存目录的大小。

按此行配置初始化后的Squid，将会在/var/spool/squid/目录下自动创建16个一级子目录(名称为00,01,02.....0F)，在每个一级子目录下再创建256个二级子目录(名称为00,01,02.....F0，F1，F2.....FF)。代理服务缓存的各种Web对象将保存在这些目录中。

➢ access_log /var/log/squid/access.log squid：指定代理服务的日志文件位置及记录，以便记录有哪些客户机通过代理访问过哪些Web对象。

➢ visible_hostname proxy.sky.com：指定代理服务器本机的可见主机名，在Squid服务的初始化或者启动过程中可能会检测此项，建议设为服务器的完整主机名(FQDN)，也可设为localhost，localdomain。

➢ dns_testnames www. google. com www. sina. com. cn www.163.com：用来执行DNS解析测试，以确保squid服务器自身的DNS查询功能正常。按从左到右的顺序，只要成功解忻出一个域名，就不再测试后边的其他域名。如果管理员确认DNS解析没问题或者不需要DNS解析，建议注释掉此项配置，以加快服务器初始化的速度。

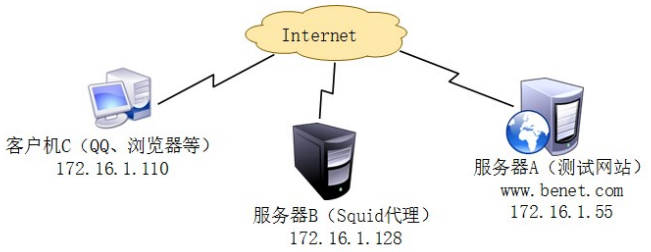
2 构建代理服务器

2.1 传统代理

使用传统代理的特点在于，客户机的相关程序(如IE浏览器，QQ聊天软件)必须指定代理服务器的地址，端口等基本信息。下面通过一个简单的应用案例来学习传统代理的配置和使用。

案例的主要需求描述如下：

- 在Linux主机B上，构建Squid为客户机访问各种网站提供代理服务，但禁止通过代理下载超过10MB大小的文件。
- 在客户机C上，指定主机B作为Web访问代理，以隐藏自己的真实IP地址。



针对上述实验环境，主机B作为代理服务器，必须正确构建Squid服务，并允许客户机使用代理；若要客户机通过代理以http：//www.sky.com的域名形式访问，则代理服务器本身必须能够正确解析该域名。主机C作为客户机，需要为浏览器等程序指定所使用的代理服务器地址，端口号等信息，主机A作为测试网站，需要启用httpd服务。

关于httpd服务器，DNS服务器的构建，请参考以前的课程，这里不再重复讲解，下面主要介绍Squid服务器的构建，客户机的代理设置，以及代理服务的验证方法.

1.Squid服务器的配置

配置Squid实现传统代理服务时，需要注意两个地方;其一，设置好可见的主机名;其二，添加http_access allow all访问策略，以便允许任意客户机使用代理服务。除此以外，为了限制下载的文件大小，还需要设置reply_body_max_size项。其他各种参数均可保持默认，也可参考1.2节做相应调整。

(1) 修改squid.conf配置文件。

```
[root@sky ~]# rpm -qa | grep squid      <==使用rpm命令检查本机是否已经安装了Porxy服务
[root@sky ~]# yum -y install squid      <==使用yum安装
[root@sky ~]# vim /etc/squid/squid.conf
http_port 3128
visible_hostname localhost.localdomain
reply_body_max_size 10240000          <==允许下载的最大文件大小(10MB)
http_access allow all                 <==放在http_access deny all之前
......  //省略部分内容
```

(2)初始化并启动squid服务。

```
[root@sky ~]# service squid start
```

第一次启动squid服务时，会自动初始化缓存目录。在没有可用的squid服务脚本的情况下，也可以直接调用squid程序来启动服务，这时需要先进行初始化。

```
[root@sky ~]# squid -z                      <==-z选项用来初始化缓存目录
[root@sky ~]# squid -D                      <==-D选项表示不进行DNS测试
```

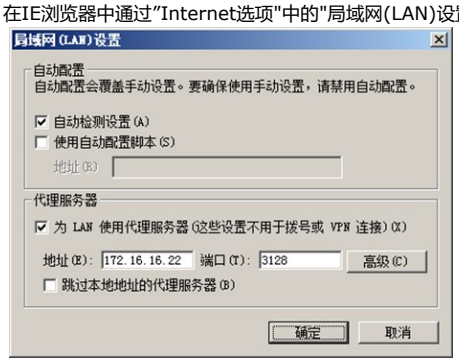
修改squid.conf配置文件以后，需要重新加载方可生效。执行"service squid reload"重新加载服务配置。

(3)确认squid服务处于正常监听状态。

```
[root@sky ~]# netstat -anpt | grep "squid"
tcp        0      0 :::3128                     :::*                        LISTEN      2763/(squid)
```

2. 客户机程序的代理配置

在IE浏览器中通过"Internet选项"中的"局域网(LAN)设置"来设置代理服务器，在Firefox浏览器中，通过"首选项"→"高级"中的"网络连接"设置代理服务器。



若要在Linux客户机的命令行界面中使用代理服务器(例如elinks网页浏览器，wget下载工具)，必须通过环境变量来指定代理服务器的地址，端口等信息。

```
[root@sky ~]# vim /etc/profile
......  //省略部分内容
HTTP_PROXY=http://172.16.1.128:3128              <==为使用HTTPS协议指定代理
HTTPS_PROXY=http://172.16.1.128:3128             <==为使用HTTPS协议指定代理
FTP_PROXY= http://172.16.1.128:3128              <==为使用FTP协议指定代理
NO_PROXY=192.168.1.,192.168.4.                   <==对两个局域网段不使用代理
export HTTP_PROXY HTTPS_PROXY FTP_PROXY NO_PROXY
```

3.验证传统代理的使用

在客户机172.16.1.110中通过浏览器访问目标网站http://172.16.1.128/，然后观察Squid代理服务器，Web服务器的访问日志，以验证代理服务是否发挥作用.

(1)查看Squid访问日志的新增记录。

在Squid代理服务器中，通过跟踪squid服务的访问日志文件，应该能够发现客户机172.16.1.110访问网站服务器172.16.1.128的记录。

```
[root@sky ~]# tail /var/log/squid/access.log
1309238261.011   34 172.16.1.110 tcp_miss/200 459 get http://172.16.1.128 /-DIRECT/172.16.1.128 text/html
1309238261.126   113 172.16.16.110 tcp_miss / 404 628 get http://172.16.1.128 / favicon. Ico-DIRECT/ 172.16.1.128 text/html
```

(2)查看Web访问日志的新增记录。

在被访问的Web服务器中,通过跟踪httpd服务的访问日志文件,应该能够发现来自代理服务器172,16,1.128的访问记录。这说明当客户机使用代理以后,Web服务器并不知道客户机的真实IP地址,因为实际上是由代理服务器在替它访问了。

```
[root@sky ~]# tail /var/log/httpd/access_log
```

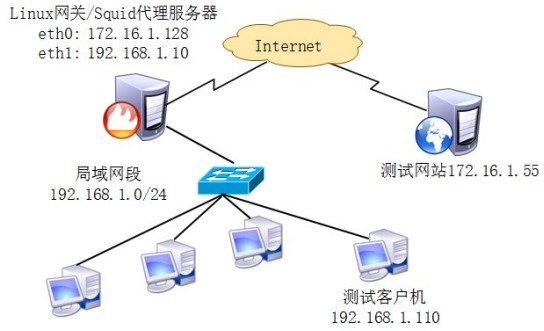
当从客户机再次访问同一Web页面时,Squid访问日志中会增加新的记录,但Web访问日志中的记录不会有变化(除非页面变更，或执行强制刷新等操作)。这说明当客户机重复访问同一静态页面时，实际上是由代理服务器通过缓存提供的。

2.2 透明代理

透明代理提供的服务功能与传统代理是一致的，但是其"透明"的实现依赖于默认路由和防火墙的重定向策略，因此更适用于为局域网主机服务，而不适合为Internet中的客户机提供服务。下面也通过一个简单的应用案例来学习透明代理的配置和使用。

案例的主要需求描述如下:

- 在Linux网关上构建Squid为客户机访问Internet提供代理服务。
- 在所有的局域网客户机上，只需正确设置IP地址，默认网关，不需要手动指定代理服务器的地址，端口等信息。



针对上述实验环境，透明代理的关键在于Linux网关服务器，而对于客户机仅需正确设置网络地址，默认网关，而并不需要指定代理服务器(若指定了反而易出错)。关于客户机的DNS解析工作，最好还是通过正常的DNS服务器来提供，不建议抛给代理服务器来处理。下面主要介绍Squid服务的透明代理设置，防火墙策略设置，其他配置操作请参考前面的传统代理构建过程。

1.配置Squid支持透明代理

Squid服务的默认配置并不支持透明代理，因此需要调整相关设置。对于2.6以上版本的Squid服务只要在http_port配置行加上一个”transparent”(透明)选项，就可以支持透明代理了。

```
[root@sky ~]# vim /etc/squid/squid.conf
http_port 192.168.1.10:3128 transparent
[root@sky ~]# service squid reload
```

2.设置iptables的重定向策略

透明代理中的Squid服务实际上是构建在Linux网关主机上面，因此只需正确设置防火墙策略，就可以将局域网主机访问Internet的数据包转交给squid进行处理。这需要用到iptables的REDIRECT(重定向)策略,其作用是实现本机端口的重新定向，将访问网站协议HTTP、HTTPS的外发数据包转交给本机的squid服务(3128端口)。

REDIRECT也是一种数据包控制类型，只能在nat表的PREROUTING或OUTPUT链以及被其调用的链中使用，通过”--to-ports 端口号"的形式来指定映射的目标端口。本例中可以将来自局域网段192.168.1.0/24且访问HTTP、HTTPS等协议的数据包，转交给运行在本机3128端口上的squid服务处理。

```
[root@sky ~]# iptables -t nat -A PREROUTING -i eth1 -s 192.168.1.0/24 -p tcp --dport 80 -j REDIRECT --to-ports 3128
[root@sky ~]# iptables -t nat -A PREROUTING -i eth1 -s 192.168.1.0/24 -p tcp --dport 443 -j REDIRECT --to-ports 3128
```

由于FTP协议涉及多个端口、多个连接，虽然也可以通过HTTP代理进行访问，但使用透明代理不便实现，因此最佳做法仍然是采用传统代理的方式——手动指定代理服务器的地址，端口号。

3.验证透明代理的使用

为了验证透明代理的效果，在客户机中应该去除手动指定的代理服务器设置(如果有的话)。例如，在IE或Firefox浏览器的连接设置中不要勾选使用代理服务器；在Linux客户机的命令行界面中，可以通过Unset命令清除HTTP_PROXY，HTTPS_PROXY等变量。

```
[root@sky ~]# unset HTTP_PROXY HTTPS_PROXY
```

在客户机192.168.1.110中通过浏览器访问目标网站http://172.16.1.128/，然后观察Squid代理服务器、Web服务器的访问日志，以验证透明代理是否发挥作用。验证结果为：在Squid代理服务器中，应该能够发现客户机192.168.1.110访问网站服务器172.16.1.128的记录；在被访问的Web服务器中，应该能够发现来自代理服务器172.16.1.128的访问记录。

2.3 ACL访问控制

Squid提供了强大的代理控制机制,通过合理设置ACL(Access Control List，访问控制列表)并进行限制，可以针对源地址、目标地址，访问的URL路径，访问的时间等各种条件进行过滤。

在配置文件squid.conf中，ACL访问控制通过以下两个步骤来实现：其一，使用acl配置定义项定义需要控制的条件；其二，通过http_access配置项对已定义的列表做“允许”或“拒绝”访问的控制。

1. 定义acl列表

每一行acl配置可以定义一条访问控制列表，格式如下所示。

acl 列表名称 列表类型 列表内容.....

其中，"列表名称"由管理员自行指定，用来识别控制条件；"列表类型"必须使用Squid预定义的值，对应不同类别的控制条件；"列表内容"是要控制的具体对象，不同类型的列表所对应的内容也不一样，可以有多个值(以空格分隔，为"或"的关系)。

通过上述格式可以发现，定义acl列表时，关键在于选择"列表类型"并设置具体的条件对象。Squid预定义的列表类型有很多种，常用的包括源地址、目标地址，访问时间，访问端口等。

常用的acl列表类型

src：源地址

dst：目标地址

port：目标端口

dstdomain：目标域

time：访问时间

maxconn：最大并发连接

url_regex：目标URL地址

Urlpath_regex：整个目标URL路径

在定义访问控制列表时，应结合当前网络环境正确分析用户的访问需求，准确定义使用代理服务的控制条件。例如，针对不同的客户机地址，需要限制访问的目标网站，特定的时间段·····，分别定义列表。

```
[root@sky ~]# vim /etc/squid/squid.conf
......  //省略部分内容
acl all src 0.0.0.0/0.0.0.0                                 <==任意客户机地址
acl localhost src 127.0.0.1/255.255.255.255                 <==源地址为127.0.0.1
acl MYLAN src 192.168.1.0/24 192.168.4.0/24                 <==客户机网段
acl to_localhost dst 127.0.0.0/8                            <==目标地址为127.0.0.0/8网段
acl MC20 maxconn 20                                         <==最大并发连接20
acl BlackURL url_regex -i ^rtsp:// ^emule://                <==以 rtsp:// 等开头的URL
acl MEDIAFILE urlpath_regex -i \.mp3$ \.mp4$ \.rmvb$        <==以.mp3、.mp4、.rmvb结尾的URL路径
acl WORKTIME time MTWHF 08:30-17:30                         <==时间为周一至周五8:30-17:30
```

当需要限制的同一类对象较多时,可以使用独立的文件来存放,在acl配置行的列表内容处指定对应的文件位置即可。例如，若要针对目标地址建立黑名单文件，可以参考以下操作。

```
[root@sky ~]# vim /etc/squid/ipblock.list
61.135.167.36
125.39.127.25
60.28.14.0/24
[root@sky ~]# vim /etc/squid/dmblock.list
.qq.com
.msn.com
.live.com
.verycd.com
[root@sky ~]# vim /etc/squid/squid.conf
acl IPBLOCK dst "/etc/squid/ipblock.list"            <==调用指定文件中的列表内容
acl DMBLOCK dstdomain "/etc/squid/dmblock.list"
```

2. 设置acl访问权限

定义好各种访问控制列表以后，需要使用httpd_access配置项来进行控制。必须注意的是，http_access配置行必须放在对应的acl配置行之后。每一行http_access配置确定一条访问控制规则，格式如下所示。

http_access allow 或 deny 列表名.....

每一条http_access规则中，可以同时包含多个acl列表名，各个列表之间以空格分隔，为"与"的关系，表示必须满足所有acl列表对应的条件才会进行限制。需要使用取反条件时，可以在acl列表前添加"!"符号。

```
[root@sky ~]# vim /etc/squid/squid.conf
......  //省略部分内容
http_access deny MYLAN MEDIAFILE          <==禁止客户机下载MP3、MP4等文件
http_access deny MYLAN IPBLOCK            <==禁止客户机访问黑名单中的IP地址
http_access deny MYLAN DMBLOCK            <==禁止客户机访问黑名单中的网站域
http_access deny MYLAN MC20               <==客户机的并发连接超过20时将被阻止
http_access allow MYLAN WORKTIME          <==允许客户机在工作时间上网
http_access deny all                      <==默认禁止所有客户机使用代理
```

执行访问控制时，Squid将按照各条规则的顺序依次进行检查，如果找到一条相匹配的规则就不再向后搜索(这点与iptables的规则匹配类似)。因此，规则的顺序安排是非常重要的，以下两种默认情况需要我们注意。

- 没有设置任何规则时：Squid服务将拒绝客户端的请求。
- 有规则但找不到相匹配的项：Squid将采用与最后一条规则相反的权限，即如果最后一条规则是allow，就拒绝客户端的请求，否则允许该请求。

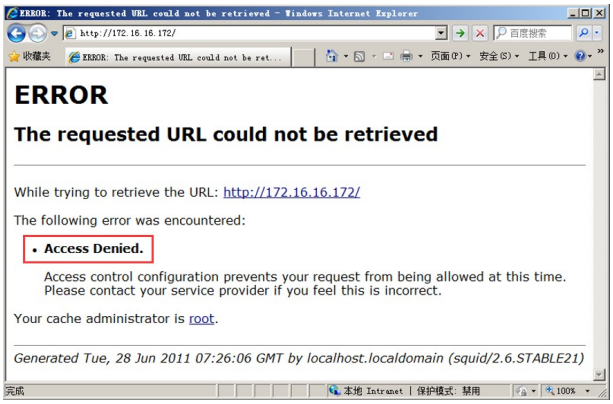
通常情况下，把最常用到的控制规则放在最前面，以减少Squid的负载。在访问控制的总体策略上，建议采用"先拒绝后允许"或"先允许后拒绝"的方式，最后一条规则设为默认策略，设为"http_access allow all"，或者"http_access deny all"。

3.验证访问控制效果

关于Squid服务的访问控制效果，无外乎两种情况：一种是能够正常访问，另一种是禁止访问。当客户机的代理访问请求被Squid服务拒绝时，在浏览器中会看到ERROR报错页面，具体内容会根据限制条件不同有些细小差别。

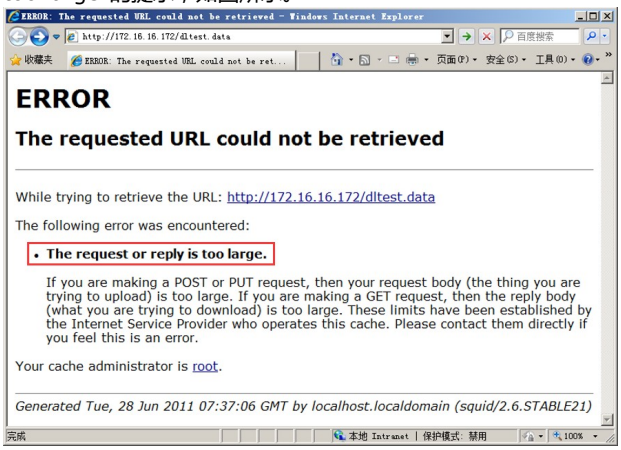
1)测试访问权限限制

对于使用http_access规则拒绝访问的情况(如访问被禁止的网站或者在禁止的时间段访问)，浏览器的报错页面中会出现"Access Denied"的提,如图所示。



2)测试文件下载限制

对于限制文件下载大小的情况(reply_body_max_size配置项),当下载超过指定大小的Web对象时，浏览器的报错页面中会出现"The request or reply is too large"的提示，如图所示。



用来下载测试的文件可以通过dd命令生成，例如若要限制大小为10MB，则可以在目标网站服务器中创建一个15MB的测试文件。

```
[root@sky ~]# dd if=/dev/zero of=/var/www/html/dltest.data bs=1M count=15
```

来自为知笔记(Wiz)
