Squid 中文权威指南

（第 4 章）

译者序：

本人在工作中维护着数台 Squid 服务器，多次参阅 Duane Wessels（他也是 Squid 的创始人）的这本书，原书名是"Squid: The Definitive Guide"，由 O'Reilly 出版。我在业余时间把它翻译成中文，希望对中文 Squid 用户有所帮助。对普通的单位上网用户，Squid 可充当代理服务器；而对 Sina,NetEase 这样的大型站点，Squid 又充当 WEB 加速器。这两个角色它都扮演得异常优秀。窗外繁星点点，开源的世界亦如这星空般美丽，而 Squid 是其中耀眼的一颗星。

对本译版有任何问题，请跟我联系，我的Email是：yonghua_peng@yahoo.com.cn

彭勇华

目 录

第 4 章 快速配置向导

4.1 squid.conf 语法

Squid 的配置文件相对规范。它与其他许多 unix 程序相似。每行以配置指令开始，后面跟着数字值或关键字。在读取配置文件时，squid 忽略空行和注释掉的行（以#开始）。如下是一些配置行示例：

```
cache_log /squid/var/cache.log
# define the localhost ACL
acl Localhost src 127.0.0.1/32
connect_timeout 2 minutes
log_fqdn on
```

某些指令取唯一值。在这些情形下，重复赋予该指令不同的值，将覆盖前面的值。例如，下面是一个连接超时值。第一行无效，因为第二行覆盖了它：

```
connect_timeout 2 minutes
connect_timeout 1 hour
```

另外，某些指令取列表值。在这些情形下，每一个新增的值都有效。"扩展方式"指令以这种方法工作：

```
extension_methods UNGET
extension_methods UNPUT
extension_methods UNPOST
```

对这些基于列表的指令，你通常能在同一行中赋予多个值：

```
extension_methods UNGET UNPUT UNPOST
```

许多指令有通用类型。例如，连接超时值是一个时间规范，在数字后面跟着时间单元。例如：

```
connect_timeout 3 hours
client_lifetime 4 days
negative_ttl 27 minutes
```

类似的，大量的指令指向文件大小或者内存额度。例如，你可以这样编写大小规范：十进制数字后面跟 bytes,KB,MB 或 GB.例如：

```
minimum_object_size 12 bytes
request_header_max_size 10 KB
maximum_object_size 187 MB
```

另一种值得提起的类型是触发器，它的值是 on 或者 off。许多指令使用该类型。例如：

```
server_persistent_connections on
strip_query_terms off
prefer_direct on
```

通常，配置文件指令能以任何顺序出现。然而，如果某个指令指向的值被其他指令所定义，那么顺序就很重要。访问控制列表是个好的例子。acl 被用在 http_access 规则之前必须被定义：

```
acl Foo src 1.2.3.4
http_access deny Foo
```

squid.conf 文件里的许多东西是大小写敏感的，例如指令名。你不能将 http_port 写成 HTTP_port。

默认的 squid.conf 文件包含了对每个指令的大量注释，以及指令的默认值。例如：

```
#  TAG: persistent_request_timeout
#       How long to wait for the next HTTP request on a persistent
#       connection after the previous request completes.
#
#Default:
# persistent_request_timeout 1 minute
```

每次安装 squid 后，当前默认配置文件存放在$prefix/etc 目录下的 squid.conf.default。既然指令每次都有所改变，你能参考该文档，以获取最近的更新。

该章剩下的部分是关于在开始运行 squid 之前，你必须知道的少数指令。

4.2 User ID

你可能知道，unix 进程和文件拥有文件和组属主的属性。你必须选择某个用户和组给 squid。该用户和组的组合，必须对大部分 squid 相关的文件和目录有读和写的权限。

我高度推荐创建名为"squid"的用户和组。这避免了某人利用 squid 来读取系统中的其他文件。假如不止一个人拥有对 squid 的管理权限，你可以将他们加到 squid 组里。

unix 进程继承了它们父进程的属主属性。那就是说，假如你以 joe 用户来启动 squid，squid 也以 joe 来运行。假如你不想以 joe 来运行 squid，你需要预先改变你的用户 ID。这是 su 命令的典型功能。例如：

```
joe% su - squid
squid% /usr/local/squid/sbin/squid
```

不幸的是，运行 squid 并非总是如此简单。在某些情况下，你必须以 root 来启动 squid，这依赖于你的配置。例如，仅仅 root 能绑定 TCP 套接字到特权端口上，如 80。假如你必须以 root 来启动 squid，你必须设置 cache_effective_user 指令。它告诉 squid，在执行完需要特别权限的任务后，变成哪个用户。例如:

```
cache_effective_user squid
```

你提供的该名字必须是有效用户（在/etc/passwd 文件里）。请注意仅仅当你以 root 来启动 squid 时，你才需要用到该指令。仅仅 root 有能力来随意改变用户身份。假如你以 joe 来启动 squid，它不能改变到 squid 用户。

你可能尝试不设置 cache_effective_user，直接以 root 来运行 squid。假如你试过，你会发现 squid 拒绝运行。这违背了安全规则。假如外部攻击者有能力危及或利用 squid，他能获取对系统的全部访问权。尽管我们努力使 squid 安全和少 bug，但还是稳重点好。

假如你没有设置 cache_effective_user,以 root 来启动 squid，squid 使用 nobody 作为默认值。不管你选择什么用户 ID，请确认它有对下面目录的读访问权：$prefix/etc,$prefix/libexec,$prefix/share.该用户 ID 也必须有对日志文件和缓存目录的写访问权。

squid 也有一个 cache_effective_group 指令，但你也许不必设置它。默认的，squid 使用 cache_effective_user 的默认组（从/etc/passwd 文件读取）。

4.3 端口号

http_port 指令告诉 squid 在哪个端口侦听 HTTP 请求。默认端口是 3128：

http_port 3128

假如你将 squid 作为加速器运行（见 15 章），你也许该将它设为 80。

你能使用附加的 http_port 行，来指示 squid 侦听在多个端口上。假如你必须支持客户组（它们被配置得不一致），这点就经常有用。例如，来自某个部门的浏览器发送请求到 3128，然而另一个部门使用 80 端口。简单的将两个端口号列举出来：

http_port 3128

http_port 8080

你也能使用 http_port 指令来使 squid 侦听在指定的接口地址上。当 squid 作为防火墙运行时，它有两个网络接口：一个内部的和一个外部的。你可能不想接受来自外部的 http 请求。为了使 squid 仅仅侦听在内部接口上，简单的将 IP 地址放在端口号前面：

http_port 192.168.1.1:3128

4.4 日志文件路径

我将在第 13 章讨论所有 squid 的日志细节。你现在你关注的唯一事情是，squid 将它的日志放在何处。默认的日志目录是 squid 安装位置下的 logs 目录。例如，假如你在./configure 时没有使用--prefix=选项，那么默认的日志文件路径是/usr/local/squid/var/logs.

你必须确认日志文件所存放的磁盘位置空间足够。在 squid 写日志时如果接受到错误，它会退出和重启。该行为的主要理由应引起你的注意。squid 想确认你不会丢失任何重要的日志信息，特别是你的系统被滥用或者被攻击时。

squid 有三个主要的日志文件：cache.log,access.log,store.log.第一个文件即 cache.log，包含状态性的和调试性的消息。当你刚开始运行 squid 时，你应密切的关注该文件。假如 squid 拒绝运行，理由也许会出现在 cache.log 文件的结尾处。在正常条件下，该文件不会变得很大。也请注意，假如你以-s 选项来运行 squid，重要的 cache.log 消息也可被送到你的 syslog 进程。通过使用 cache_log 指令，你可以改变该日志文件的路径：

cache_log /squid/logs/cache.log

access.log 文件包含了对 squid 发起的每个客户请求的单一行。每行平均约 150 个字节。也就是说，在接受一百万条客户请求后，它的体积约是 150M。请使用 cache_access_log 指令来改变该日志文件的路径：

cache_access_log /squid/logs/access.log

假如因为某些理由，你不想 squid 记录客户端请求日志，你能指定日志文件的路径为 /dev/null.

store.log 文件对大多数 cache 管理员来说并非很有用。它包含了进入和离开缓存的每个目标的记录。平均记录大小典型的是 175-200 字节。然而，squid 不在 store.log 里对 cache 点击创建接口，所以它比 access.log 包含少得多的记录。请使用 cache_store_log 指令来改变它的位置：

cache_store_log /squid/logs/store.log

通过指定路径为 none，你能轻易的完全禁止 store.log 日志：

cache_store_log none

假如你不小心，squid 的日志文件增加没有限制。某些操作系统对单个文件强制执行 2G

的大小限制，即使你有充足的磁盘空间。超过该限制会导致写错误，这样 squid 就会退出。为了保证日志文件大小合理，你应创建任务来有规律的重命名和打包日志。squid 有内建功能来使这个容易做到。请见 13.7 章关于日志轮循的解释。

4.5 访问控制

在第 6 章里有更多的关于访问控制的描述。现在，我只讲述少量的访问控制方法，以使热心的读者能快速开始使用 squid。

squid 默认的配置文件拒绝每一个客户请求。在任何人能使用代理之前，你必须在 squid.conf 文件里加入附加的访问控制规则。最简单的方法就是定义一个针对客户 IP 地址的 ACL 和一个访问规则，告诉 squid 允许来自这些地址的 HTTP 请求。squid 有许多不同的 ACL 类型。src 类型匹配客户 IP 地址，squid 会针对客户 HTTP 请求检查 http_access 规则。这样，你需要增加两行：

```
acl MyNetwork src 192.168.0.0/16
http_access allow MyNetwork
```

请将这些行放在正确的位置。http_access 的顺序非常重要，但是 acl 行的顺序你不必介意。你也该注意默认的配置文件包含了一些重要的访问控制，你不应该改变或删除它们，除非你完全理解它们的意义。在你第一次编辑 squid.conf 文件时，请看如下注释：

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
```

在该注释之后，以及"http_access deny all"之前插入你自己的新规则。

为了彻底说明，如下是一个合理的初始访问控制配置，包括推荐的默认控制和早先的例子：

```
acl All src 0/0
acl Manager proto cache_object
acl Localhost src 127.0.0.1/32
acl Safe_ports port 80 21 443 563 70 210 280 488 591 777 1025-65535
acl SSL_ports 443 563
acl CONNECT method CONNECT
acl MyNetwork src 192.168.0.0/16

http_access allow Manager Localhost
http_access deny Manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow MyNetwork
http_access deny All
```

4.6 可见主机名

希望你不必担心 visible_hostname 指令。然而，假如 squid 不能发现它所运行的机器的主机名，你就必须设置它。如果发生这样的事，squid 抱怨和拒绝运行：

```
% squid -Nd1
FATAL: Could not determine fully qualified hostname.   Please set 'visible_hostname'
```

有大量的理由使 squid 需要知道主机名：

主机名出现在 squid 的错误消息里，这帮助用户验证潜在问题的源头。

主机名出现在 squid 转发的 cache 单元的 HTTP Via 头里。当请求到达原始主机时，Via 头包含了在传输过程中涉及的代理列表。squid 也使用 Via 头来检测转发环路。我将在第 10 章里讨论转发环路。

squid 对特定事务使用内部 URL，例如 FTP 目录列表的图标。当 squid 对 FTP 目录产生 HTML 页面时，它插入小图标用以指明该目录中的文件类型。图标 URL 包含了 cache 的主机名，以便 web 浏览器能直接从 squid 请求它们。

每个从 squid 响应的 HTTP 回复包含了 X-Cache 头。这并非官方 HTTP 头。它是一个扩展头，用以指明该响应是 cache 点击还是 cache 丢失。既然请求和响应可能经过多个 cache，每个 X-Cache 头包含了 cache 报告点击或丢失的名字。如下是一个通过 2 个 cache 的响应示例：

```
HTTP/1.0 200 OK
Date: Mon, 29 Sep 2003 22:57:23 GMT
Content-type: text/html
Content-length: 733
X-Cache: HIT from bo2.us.ircache.net
X-Cache: MISS from bo1.us.ircache.net
```

squid 在启动时试图自动获取主机名。首先它调用 gethostname()函数，这通常能返回正确的主机名。接着，squid 调用 gethostbyname()函数尝试对主机名进行 DNS 查询。该函数典型的返回 IP 地址和系统的规范名。假如 gethostbyname()成功，squid 在错误消息里，Via 头里等地方使用这个规范名。

因为大量的理由，squid 可能不能检测到它的规范主机名，包括：

主机名可能未设置。

主机名可能从 DNS 区域或/etc/hosts 文件里丢失。

squid 系统的 DNS 客户端配置可能不正确或丢失。在 unix 系统上，你该检查/etc/resolv.conf 和/etc/host.conf 文件。

假如你看到上述的致命错误，你必须修正主机名和 DNS 信息，或者显式的给 squid 指明主机名。在大多数情况下，请确认"hostname"命令返回一个完全规范的主机名，并且在/etc/hosts 文件里增加这个接口。假如这样不成功，请在 squid.conf 里设置可见主机名：

```
visible_hostname squid.packet-pushers.net
```

4.7 管理联系信息

你应该设置 cache_mgr 指令作为对用户的帮助。它是一个 email 地址，假如问题发生，用户能写信给它。cache_mgr 地址默认出现在 squid 的错误消息里。例如：

```
cache_mgr squid@web-cache.net
```

4.8 下一步

在创建了初步的配置文件后，你多少准备首次运行 squid 了。请遵循下面章节的建议。

当你掌握了启动和停止 squid 后，你该花费一些时间来改善配置文件。你可能想增加更

高级的访问控制，这在第 6 章里有描述。既然我在这里没有讨论磁盘 cache，你该花些时间阅读第 7 和第 8 章。