China-pub.com
下载

China-pub.com
下载

第5章　基本Web服务

Apache Web服务器是目前在Internet上最为广泛使用的Web服务器平台，而且它已经被移植成了操作系统的一部分。与其他商用的 Web服务器相比较， Apache的最大竞争优势在于它的源代码是完全开放的。正因为 Apache的开放性设计，许多功能强大的模块被开发出来，极大地扩展了Apache的功能。

Apache是Red Hat Linux 6.0的一个主要组成部分，但是Linux下Web服务器的配置和标准Apache的安装存在较大的区别，正因为如此，一些人把这种配置方式称为“ Red Hat 方式”。

5.1　基本Apache配置

若Apache是由源代码编译而来，那么它通常被安装到 /usr/local/etc/httpd目录中，在Red Hat Linux安装时是不会创建该目录的。

下面是对Red Hat Linux下的各种Apache文件和目录的描述：

1. /etc/httpd/conf

这个目录包含了下列三个重要的 Apache配置文件：

httpd.conf：主要的Web服务器配置。

access.conf：访问限制和安全。

srm.conf：MIME与文件关联。

2. /var/log/httpd

这个目录包含了下面一些Web服务器的日志文件：

access_log：记录所有与Web服务器的连接

error_log：报告运行时服务器所遇到的所有错误（如 CGI、.htaccess配置错误等）。

日志处理系统将定期地为日志改名或删除。

3. 其他目录和文件

/etc/mimes.types：Apache使用该文件将文件类型与文件扩展名相关联。

/etc/rc.d/init.d/httpd：这是一个Shell脚本文件，用来启动和关闭 Web服务器。在调用时应带start或stop参数（若不带参数执行此脚本文件，则将产生一个提示调用方法的信息）。

/usr/sbin/httpd：这是存放Web服务器可执行文件的目录。这个目录还包含一些 Web服务器的支撑工具（如本章后面讨论的 htpasswd）。

/home/httpd/html：这是缺省的文档根目录。在本目录下的文件和目录是 Web可访问的。当使用Web浏览器与Apcache服务器连接时，该目录下的 index.html文件是缺省的被浏览文件。

/home/httpd/cgi-bin：这个目录包含了支持 Web站点的一些可执行代码文件。这些可执行文件必须遵守CGI的规范，但是可以用很多种语言来编写，包括已编译的 C/C++代码、Java、Perl、PHP、BASH、awk或其他已编译的代码或解释器。这些文件的可执行用户为 nobody，所以需要特别关心权限问题，否则CGI实现可能会带来严重的安全性问题。

/home/luser/public_html：系统中的每一个用户在 Web服务器中都可以有自己的空间，这

China-pub.com
下载

是通过创建public_html目录并使之可浏览来实现的。例如：本地用户 luser可以使用如下的命令来创建自己的Web空间：

```
cd ~
chmod o+x ~
mkdir public_html
echo '<b>Sample HTML file</b>' > public_html/index.html
```

这个目录下文件的内容，可以通过 URL http：//127.0.0.1/~luser/index.html来访问Web服务器获得（在使用时，可以用正式域名来取代 localhost）。

5.2 使用和不使用IP别名的虚拟主机

5.2.1 使用IP别名的虚拟主机

假定Linux系统中的eth0以太网接口已经配置了 IP地址1.2.3.3，那么下面的命令可以使得eth0对IP地址1.2.3.4也同样作出响应（这便是建立虚拟主机的中心思想）：

```
/sbin/ifconfig eth0:1 1.2.3.4
/sbin/route add -host 1.2.3.4 dev eth0:1
```

显然，在使用这两条命令之前必须确认网络中的其他主机没有使用 IP地址1.2.3.4。当用户对此不能确定的时候，应该与网络管理员联系，而不要去盲目猜测。

应该把这些命令加入到系统的启动脚本中，但是要注意的是，它们应该在主要的 ifconfig命令之后执行（最好是放置在文件 /etc/rc.d/init.d/network中）。

虚拟接口的次序如下：从eth0:1开始，继之以eth0:2、eth0:3、eth0:4……，这样各个独立的IP地址就与对应的虚拟接口相关联。在一个网络接口上允许的最大虚拟 IP地址数为255。

一旦在网络接口上添加了新的 IP地址，对应地应该加入一条新的 DNS入口记录，该记录中包含有指向新入口的地址。

Web服务器必须为每一个 IP地址配置不同的文档根目录。在 Apache服务器中通过为每一个接口在/etc/httpd/conf/httpd.conf文件的结尾处加入清单 5-1所示的一段代码，就可以实现上述动作的初始化过程。

清单5-1 虚拟主机

```
<VirtualHost 1.2.3.4>
ServerAdmin webmaster@luser.com
DocumentRoot /home/luser/public_html
ServerName www.luser.com
ErrorLog logs/luser-error_log
TransferLog logs/luser-access_log
RefererLog logs/luser-referer_log
AgentLog logs/luser-agent_log
</VirtualHost>
```

为了真实反映用户的 Web站点配置，上述清单中的一些行必须更改。通常需要被定制的内容包括：文档根目录、服务器名称和各种日志文件。

Apache的文档中建议对上述代码中的 VirtualHost指令使用 IP地址，而不是主机名。若Apache服务器启动时， DNS服务器因为某种原因不可用，则 VirtualHost指令解析失败，从而引发整段代码被忽略而没有执行。使用 IP地址就可以避免这种潜在的错误。

China-pub.com 下载

关于Linux下虚拟主机的完整描述可以在IP-Alias-HOWTO中找到。该文件在本书所附光盘的HOWTO RPM软件包中，还可以通过一些在线的Linux文档资源（如http://www/linuxhq.com）得到。

5.2.2 基于名称的虚拟主机

HTTP/1.1协议支持以下的机制： Web服务器可以获得访问它的浏览器所在主机的名称。基于这些名称， Apache可以被配置成能提供多个 Web站点服务的形式，即使 DNS将这些站点都解析到同一个IP地址。

如清单5-2所示，基于名称的虚拟主机配置与基于 IP别名的虚拟主机配置区别极小（唯一的区别在于NameVirtualHost指令）。

清单5-2 名称虚拟主机

```
NameVirtualHost 1.2.3.4

<VirtualHost 1.2.3.4>
ServerAdmin webmaster@luser.com
DocumentRoot /home/luser/public_html
ServerName www.luser.com
ErrorLog logs/luser-error_log
TransferLog logs/luser-access_log
RefererLog logs/luser-referer_log
AgentLog logs/luser-agent_log
</VirtualHost>
```

为了适应不支持 HTTP/1.1协议的一些旧版本浏览器（如 Windows NT4中附带的 MS Explore2）的需要，还可以使用 ServerPath指令。ServerPath指令以一个目录作为参数，该目录与文档根目录对应。当在这个目录下的文件被用户请求时，该文件被假定属于用户所在的主机。

5.2.3 使用JavaScript的虚拟主机

建立虚拟主机的最好方法是将服务器的网卡配置多个网络地址，因为这项技术在所有浏览器下均能正常地工作。另一种可供选择的方法是通过使用 JavaScript驱动的浏览器来实现的。

假设一台服务器的网卡只配置了一个 IP地址，而且域名解析将名字 www.company-a.com和www.company-b.com都解析成该IP地址；又假设在该 Web服务器的文档根目录下存在一个index.html文件，服务器文档根目录应该对这两个不同的公司提供不同的页面。

JavaScript中的Location变量包含了用于访问当前页面的 URL，此外若Location被修改，浏览器将立刻载入变量中包含的新 URL。

利用这个特点， JavaScript可以用下面这种简单的脚本实现虚拟主机：

```
<script language="JavaScript">
<!--
if (location=="http://www.company-b.com/")
        location="http://www.company-a.com/company-b/"
//-->
</script>
```

China-pub.com
下载

这样，浏览器（而不是服务器）可以决定调用哪一个页面。

JavaScript会在一定范围内对Location变量进行修改，以确保它包含http://和结尾的斜线。这种方法对Web管理员是十分方便的，因为他们通常没有对整个机器的全部管理权限；同时对某一个文件实施一定的改变相对于将网络接口重新进行配置要容易得多。

这种方法的最大缺点在于不使用JavaScript的浏览器（如Lynx）将不执行脚本和调用新的页面，同时即便使用JavaScript的浏览器（如Netscape2和Internet Explorer 3等），在用户关闭JavaScript之后也会出现这种错误。

另外，这种方法还有以下一些问题：

- 在某种程度上，浏览器将在页面的位置改变之前给出前一页面的 HTML；为了减少这一影响，可以将JavaScript的代码段尽量靠近开始时的<HEAD>标记。
- 浏览历史列表包含的是原始的URL，而不是重新定向的位置。
- 当用户点击浏览器上的“ Back ”按钮时，将重新加载原始的URL。原始的URL立即被重新定向，这使得“ Back ”按钮的作用完全失效。

JavaScript的replace( )方法则提供了较大的灵活性。在replace( )方法被执行以后，用户并不能通过点击“ Back ”按钮去浏览前一个 URL。但是Netscape Navigator版本2并不支持replace( )，因此必须执行相应的版本测试。清单5-3就是使用replace( )的一个示例。

清单5-3 使用JavaScript的虚拟主机

```
<script language="JavaScript">
<!--
var js_ok = false;
if (navigator.appName.substring(0,9) == "Microsoft" &&
        parseInt(navigator.appVersion) >= 4 )
        js_ok = true;
if (navigator.appName.substring(0,8) == "Netscape" &&
        parseInt(navigator.appVersion) >= 3 )
        js_ok = true;

   if (location=="http://www.company-b.com/")
        if (js_ok == true)
             replace("http://www.company-a.com/company-b/")
        else
             location="http://www.company-a.com/company-b/"
//-->
</script>
```

5.3 静态口令保护

对Web页面实施口令保护是一件相对容易的事。这种简单的 Web安全形式只需要创建或修改Web服务器中的一小部分文件。

我们用.htaccess文件来配置这种基本的安全特征：在这个配置文件中的设置将应用到位于文档根目录及其子目录下的所有HTML文件。

了解使用这种保护方法的局限性是非常重要的，它并不十分安全。事实上它还很容易受到攻击，因此不要相信使用.htaccess文件就可以保护很重要的数据。

使用.htaccess文件保护的Web页面必须是能够被Web服务器中的某个用户（通常是nobody

China-pub.com 下载

用户， nobody用户组）所读取的。除非Web页面的作者将Web服务器及其页面配置为组内共享的方式（假定服务器运行在标准端口上，这个配置操作在没有超级用户的帮助下是无法进行的），页面必须具有任何人均可读的权限才能使得它在服务器上可以被访问。若设置了这样的权限，那么通过shell访问系统的其他用户都可以读取页面，而不用考虑 .htaccess文件的存在限制；更进一步， CGI应用程序（尤其是将在后面讨论的 PHP）将具有访问该全局可读数据的权限。对标准Red Hat Web服务器而言，如果Web服务器进程可以读取一个数据，那么系统中的任何用户都可以读取该数据。如果读者对上述讨论不是很清楚的话，那么只需要简单地认识到：Web驻留程序提供服务的数据不是安全的，除非系统中没有任何其他用户。

另外，用户应该清楚地记得，所有 HTTP数据将以清晰的文本形式传送到中间网络。除非配置了安全套接字层（ Secure Socket Layer， SSL），否则数据将不进行任何加密操作。这样数据将被网络中的数据包探测器（如 sniffit）所截获，或者驻留在代理的缓存中。

牢记上面提到的这些警告，下面将简单地讨论访问控制的方法。

Red Hat Linux 6.0中缺省的Apache配置是不允许使用.htaccess文件的；为了启用文件使用认证，必须对文件/etc/httpd/conf/access.conf进行一定的修改。这种修改实施在文件 <Directory /home/httpd/html>段内以AllowOverride开始并以一系列选项结束的那一行。这一行最低限度应该包含如下形式：

```
AllowOverride AuthConfig Limit
```

除了上述定义的权限，关键字 All可用于对.htaccess文件访问的进一步控制；但是如果没有功能上的需要，那么就不要使用 All设置激活其余的特征。

在缺省的/etc/httpd/conf/access.conf文件中， AllowOverride语句将出现三次：一次是针对全局选项，一次是针对 /cgi-bin目录下的执行程序，一次是针对在文档根目录下的所有页面。每一段都使用<Directory>标记封装起来，只需要针对需要访问控制的段进行修改即可。

在进行了修改以后， Web服务器必须重启。Web服务器的重启可以通过向主Web服务器发HUP信号完成，也可以利用rc文件来进行控制：

```
/etc/rc.d/init.d/httpd restart
```

当Web服务器通过HUP或重新启动来重新读取 Web服务器的配置以后，这些安全特征将被启用。为了测试这些特征，系统管理员可以使用超级用户的帐号执行下列命令来建立一个Web页面的示例：

```
mkdir /home/httpd/html/restrict
echo "<hr>HTML Rules<hr>" >
/home/httpd/html/restrict/test.html
```

该文件可以在本地通过URL http://127.0.0.1/restrict/test.html来访问（显然，如果是外部浏览器访问，则需要一个限定的主机名）。

为了保护该文件以及这个目录下的所有文件，应该创建一个内容如清单 5-4所示的/home/httpd/html/restrict/.htaccess文件。

清单5-4 使用.htaccess的访问控制

```
deny from all
allow from permitted.com
AuthType Basic
AuthUserFile /etc/httpd/htpasswd
```

China-pub.com
下载


```
AuthName "special directory"
require valid-user
satisfy any
```

在这个文件创建后，键入如下命令：

```
/usr/sbin/htpasswd -c /etc/httpd/htpasswd bogususer
```

系统将要求提供口令，输入一个口令并加以确认；现在再检测 /etc/httpd/htpasswd文件，它应该和如下内容类似：

```
bogususer:TPNWusI5zZdqs
```

在正确地完成上述各个步骤以后，使用浏览器来访问 URL http://127.0.0.1/restrict/test.html就需要提供用户名和口令了。上面的文件对于来自 permitted.com（需要逆向域名解析服务器正确地工作）的访问请求将不作任何限制，当然也可以在该位置使用 IP地址。如果允许任意地访问，那么只需要简单地将清单 5-4中的allow行全部删除即可。

htpasswd文件可以管理多个帐号和口令。为了增加一个新的用户及其口令，可以键入如下的命令：

```
/usr/bin/htpasswd /etc/httpd/htpasswd realuser
```

注意，与前一个htpasswd命令相比，该命令缺少一个“ -c ”任选项。这是因为文件htpasswd已经存在了，不需要再次创建。在为新用户键入口令之后， /etc/httpd/htpasswd将包含新的一行：

```
bogususer:TPNWusI5zZdqs
realuser:Mv19hxh2ff07k
```

拥有独立口令的多个用户可以使用这种方法加入。Apache的文档中有这样的警告：htpasswd文件增长过多的话将导致访问速度变慢。将 htpasswd中的数据移到数据库中可以解决大量的授权用户列表问题。 Apache文档中推荐使用 DBM文件， Sybase数据库的授权问题将在本书第6章讨论。

最后需要注意的是，聪明的用户可能注意到 htpasswd文件中的表项与包含所有用户信息的/etc/passwd数据库（或在执行了 pwconv工具后的/etc/shadow文件）表项很相似。实际上，在这两个文件中使用了相同的加密技术（ DES，而不是MD5），因此将口令的乱码自由地从一个文件移到另一个文件是完全可行的。

一个没有经验的系统管理员可能试图去将 /etc/passwd文件和 htpasswd文件同步，从.htaccess文件中直接调用/etc/passwd文件。要小心的是，这样的配置会给攻击者打开一道方便出入的大门，因此在实际操作中最好不要这样。