



### 问题： 怎样用 webalizer 分析 web 日志？

#### 解决方法：

webalizer 是一个高效的、免费的 web 服务器日志分析程序。其分析结果以 HTML 文件格式保存，从而可以很方便的通过 web 服务器进行浏览。Internet 上的很多站点都使用 webalizer 进行 web 服务器日志分析。Webalizer 具有以下一些特性：

1. 为是用 C 写的程序，所以其具有很高的运行效率。在主频为 200Mhz 的机器上，webalizer 每秒钟可以分析 10000 条记录，所以分析一个 40M 大小的日志文件只需要 15 秒。

2. webalizer 支持标准的一般日志文件格式(Common Logfile Format)；除此之外，也支持几种组合日志格式(Combined Logfile Format)的变种，从而可以统计客户情况以及客户操作系统类型。并且现在 webalizer 已经可以支持 wu-ftpd xferlog 日志格式以及 squid 日志文件格式了。

3. 支持命令行配置以及配置文件。

4. 可以支持多种语言，也可以自己进行本地化工作。

5. 支持多种平台，比如 UNIX、linux、NT, OS/2 和 MacOS 等。

#### 安装：

1. 从 webalizer 的官方网站 <http://www.mrunix.net/webalizer/> 下载 webalizer ,当前的最新版本是 webalizer-2.01-06-src.tgz。

2. 首先解开源代码包：

```
tar xvzf webalizer-2.01-06-src.tgz
```

3. 在生成的目录中有个 lang 目录，该目录中保存了各种语言文件，但是只有繁体中文版本，可以自己转换成简体，或者自己重新翻译一下。

4. 然后进入生成的目录：

```
./configure  
make --with-language=chinese
```

5. 编译成功后，会产生一个 webalizer 可执行文件，可以将其拷贝到/usr/sbin/目录下：

```
cp webalizer /usr/sbin/
```





然后就可以开始配置 `webalizer` 了。

配置：

上面说过，可以通过命令行配置 `webalizer`，也可以通过配置文件进行配置，在本文中我们将介绍使用命令行参数进行配置，需要了解配置文件使用方法的朋友可以参考 `README` 文件，里面有很详细的介绍。

可以执行 `webalizer -h` 得到所有命令行参数：

```
Usage: webalizer [options] [log file]
-h = 打印帮助信息
-v -V = 打印版本信息
-d = 打印附加调试信息
-F type = 日志格式类型. type= (clf | ftp | squid)
-i = 忽略历史文件
-p = 保留状态 (递增模式)
-q = 忽略消息信息
-Q = 忽略所有信息
-Y = 忽略国家图形
-G = 忽略小时统计图形
-H = 忽略小时统计信息
-L = 忽略彩色图例
-l num = 在图形中使用数字背景线
-m num = 访问超时 (seconds)
-T = 打印时间信息
-c file = 指定配置文件
-n name = 使用的主机名
-o dir = 结果输出目录
-t name = 指定报告题目上的主机名
-a name = 隐藏用户代理名称
-r name = 隐藏访问链接
-s name = 隐藏客户
-u name = 隐藏 URL
-x name = 使用文件扩展名
-P name = 页面类型扩展名
-l name = index 别名
-A num = 显示前几名客户类型
-C num = 显示前几名国家
-R num = 显示前几名链接
```

地址：北京市海淀区万泉河路 68 号 紫金大厦 6 层  
邮编：100086





- S num = 显示前几名客户
- U num = 显示前几名 URLs
- e num = 显示前几名访问页面
- E num = 显示前几名不存在的页面
- X = 隐藏个别用户
- D name = 使用 dns 缓存文件
- N num = DNS 进程数 (0=禁用 dns)

假设，web 服务器主机名为 www.test.com，统计站点域名为 www.test.com，访问日志为/var/log/httpd/access\_log，我们将 webalizer 分析结果输出到/var/www/html/log 下面。则我们可以建立以下脚本/etc/rc.d/webalizer：

```
#!/bin/sh
run=/usr/sbin/webalizer
$run -F clf -p -n ' ' -t 'www.test.com'
-o /var/www/html/log /var/log/httpd/access_log
```

说明：

-F clf 指明我们的 web 日志格式为标准的一般日志文件格式(Common Logfile Format)

-p 指定使用递增模式，这就是说每作一次分析后，webalizer 会生产一个历史文件，这样下一次分析时就可以不分析已经处理过的部分。这样我们就可以在短时间内转换我们的日志文件，而不用担心访问量太大时日志文件无限增大了。

-n “ ” 指定服务器主机名为空，这样输出结果会美观一些。

-o “www.test.com” 指定输出结果标题。

/var/log/httpd/access\_log: 指定日志文件

然后在/etc/crontab 中加入：

```
01 1 * * * root /etc/rc.d/webalizer
```

即每天凌晨 1 点执行该脚本。

然后运行/etc/rc.d/init.d/crond reload 重载入 crond 服务。

测试：

执行以下命令：

地址：北京市海淀区万泉河路 68 号 紫金大厦 6 层  
邮编：100086





```
# /etc/rc.d/webalizer
```

然后在浏览器中访问 <http://www.test.com/log/> 就可以看到 webalizer 的分析结果了。

注意：如果您使用了中文语言文件，但是您的 Linux 不支持中文，则在产生的图片中文字可能为乱码。

### 问题： 如何在一个系统上编译多个核心版本的驱动模块？

解决方法：

我们可以只在一个系统中使用一份核心源代码就编译出来不同的驱动模块，方法如下：

- 1、安装相应版本的核心源代码。
- 2、进入 /usr/src 目录，假设你安装的是 2.4.17-1 的源代码，那么需要建立联结 linux 指向 /usr/src/linux-2.4.17-1 目录，因为许多驱动源代码都需要使用 /usr/src/linux 目录。
- 3、检查你系统中下面两个目录：

```
/usr/include/linux
```

```
/usr/include/asm
```

如果不存在或不是联结，那么需要建立联结，指向 /usr/src/linux/include/linux 和 /usr/src/linux/include/asm：

```
ln -s /usr/src/linux/include/linux /usr/include/linux
```

```
ln -s /usr/src/linux/include/asm /usr/include/asm
```

- 4、下面开始修改核心源代码的配置以适应编译不同版本的驱动模块：

\*首先进入 /usr/src/linux 目录，执行命令 `make mrproper`，然后确定你需要编译的版本，假设是 2.4.17-1B00T，那么首先进入 /usr/src/linux 目录，修改 Makefile 文件，文件前几行如下：

```
VERSION = 2
```

```
PATCHLEVEL = 4
```

```
SUBLEVEL = 17
```

```
EXTRAVERSION = -1custom
```

将最后一行改为你需要的 -1B00T，然后保存退出。

然后将 `configs/kernel-2.4.18-i386-B00T.config` 文件拷贝到当前目录下，命名为 `.config`，然后执行命令 `make oldconfig` 进行配置，完成后执行命令 `make dep`，之后就完成了核心源代码的设置工作。

如果你需要编译其他版本，比如 SMP 的，那么需要重复上面几个步骤，将 Makefile 中那行改为 `-1smp`，然后将 `configs/kernel-2.4.18-i686-smp.config` 文件拷贝到当前目录下命名为 `.config`，然后也是连续执行 `make oldconfig` 和 `make dep` 命令，之后就可以重新编译你的驱动源代码了。

- 5、编译驱动源代码时可以参考其中的 README 或 INSTALL 文件，修改还核心源代码配置后就可以开始编译了，编译好一个版本之后一定要记得备份，因为下次编译会冲掉原来的驱动模块。驱动模块编译好之后可以通过查找其中的关键字 `kernel_version` 来查看相应的版本。





6、如果你只是拿到了几个 C 文件和头文件，说明中说要替换掉核心源代码中相应的文件然后重新编译核心模块，那么就会需要很长的编译时间，这里有一个简单的方法，就是只编译这个驱动而不需要重新编译所有核心模块，使用命令：

```
gcc -DMODULE -D__KERNEL__ -O6 -c filename.c
```

可以直接将 C 程序编译成驱动模块，在当前目录下生成。

\*如果有多个 C 程序，可以分别使用上述命令编译，然后使用命令 `ld -r -o destname.o sourcename.o` 进行连接就可以了。

7、如果是需要放在核心源代码中编译的，可以执行这个命令：`make -n modules > cmd.sh` 这个命令不是编译模块，只是将编译时要执行的命令打出来，所以我们可以编辑 `cmd.sh` 文件，

找到编译你的那些模块的命令，然后将他们拷贝出来，另存为一个脚本文件，然后在相应的目录下执行你的脚本文件就可以得到驱动模块了。

**问题：怎样配置 SSH, sftp, telnet 等服务？**

解决方法：

ssh 服务和 telnet 服务都是远程登录控制服务器，不能实现上传下载；

开启 ssh 服务的方法是在终端执行：

```
/etc/init.d/sshd start
```

关闭执行：

```
/etc/init.d/sshd stop
```

配置打开 telnet 服务

在终端执行：

```
vi /etc/xinetd.d/telnet
```

打开文件后，将里面的 `disable=yes` 行前面加上 # 注释掉，保存退出，重新启动 `xinetd` 服务

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

注意，telnet 不允许 root 用户直接登录，需要您先建立一个普通用户，使用这个用户登录，再转换到 root 用户。

限制 proftpd 的上传下载速率：

在 `/etc/proftpd.conf` 文件中加上

```
TransferRate RETR 50 # 下载速率限制在 50Kbytes/s
```

```
TransferRate STOR 100 # 上传的速率控制在 100Kbytes/s
```

sftp 服务的配置文件是二进制文件，不可修改，也不能限制用户上传下载，但是您可以将 sftp 服务禁止，这样其他用户就不能使用 sftp 服务，禁止方法是执行：

```
vi /etc/ssh/sshd_config
```

将文件中最后一行调用 sftp 的语句前面加上 # 注释掉，保存退出，重新启动 ssh 服务即可，此时 scp 也不能使用了。

**问题： 怎么用 tar 命令简单实现数据的全备份、增量备份、增量备份呢？**





解决方法：

命令基本格式是：

```
tar cvzf TARGET SOURCES -N TIME
```

e.g

```
tar cvzf foo.tgz /bak -N "2004-03-03 16:49:17"
```

记住全备份的时间 `f_time` 和上一次增量备份的时间 `i_time`; (现在的办法是以 `job` 为单位记录这两个时间: " [JobName] [F\_TIME] [I\_TIME] '\n' ")

全备份: `tar cvzf foo.tgz /bak`

增量备份: `tar cvzf foo.tgz /bak -N i_time`

差量备份: `tar cvzf foo.tgz /bak -N f_tim`

**问题: 如何修改网卡的 Mac 地址?**

解决方法：

首先必须关闭网卡 `eth0`, 否则会报告系统忙, 无法更改。命令是：

```
ifconfig eth0 down
```

```
ifconfig eth0 hw ether 00:AA:BB:CC:DD:EE
```

```
ifconfig eth0 up
```

网卡的 MAC 地址更改就完成了。

如果想下次启动的时候, 保持原来的设置, 可以在 `/etc/rc.d/rc.sysinit` 文件中加入命令:

```
ifconfig eth0 down
```

```
ifconfig eth0 hw ether 00:AA:BB:CC:DD:EE
```

```
ifconfig eth0 up
```

这个脚本运行在 `network` 之前, 所以 MAC 跟 IP 就是对应的了。

