目录

[DIR 2](#_Toc475699003)

[> ,>> 重定向 2](#_Toc475699004)

[if exist 2](#_Toc475699005)

[netstat 检查所有的网络端口状态 2](#_Toc475699006)

[：： 注示 3](#_Toc475699007)

[cls 清屏 3](#_Toc475699008)

[示例 3](#_Toc475699009)

[Echo反馈 4](#_Toc475699010)

[@ 命令 - 不显示@以后的命令并执行 5](#_Toc475699011)

[autoset 当需要确认时，效果和/y是一样 5](#_Toc475699012)

[:和goto 5](#_Toc475699013)

[Rem注释命令 5](#_Toc475699014)

[Pause暂停命令 6](#_Toc475699015)

[Call 命令 6](#_Toc475699016)

[start 调用外部程序 6](#_Toc475699017)

[choice用户输入一个字符，从而运行不同的命令 7](#_Toc475699018)

[If 命令 7](#_Toc475699019)

[errorlevel 执行命令返回码，表示是否出错 8](#_Toc475699020)

[find搜索命令 8](#_Toc475699021)

[type命令列出a.txt中的内容 8](#_Toc475699022)

[参数%命令 8](#_Toc475699023)

[| 通道命令 9](#_Toc475699024)

[|| 通道命令 9](#_Toc475699025)

[& 通道命令 9](#_Toc475699026)

[&& 通道命令 9](#_Toc475699027)

## DIR

dir d:\work /ad /s /b 最常用的命令：列出某目录下所有子文件夹

dir d:\work\\*\_??.jpg /a-d /s /b

最常用的命令：列出某目录下包括子文件夹内的所有满足条件的文件

例: asdddf\_89.jpg ; 5554\_aa.jpg

/ad :a是设置，d是文件夹，不设置此项，则会列出文件

/a-d “ - ”是”非”，就是要求列出 文件夹 以外的所有

/s 包括子目录，默认不包括

/b 排除文件名外的附加信息

## > ,>> 重定向

ping sz.tencent.com > a.txt

ping sz1.tencent.com >> a.txt

ping sz2.tencent.com >> a.txt

ping sz3.tencent.com >> a.txt

ping sz4.tencent.com >> a.txt

ping sz5.tencent.com >> a.txt

ping sz6.tencent.com >> a.txt

ping sz7.tencent.com >> a.txt

exit

这里>的意思，是把前面命令得到的东西放到后面所给的地方，>>的作用，和>的相同，区别是把结果追加到前一行得出的结果的后面，具体的说是下一行，而前面一行命令得出的结果将保留，这样可以使这个a.txt文件越来越大（想到如何搞破坏了？？）。By the way，这个批处理还可以和其他命令结合，搞成完全自动化判断服务器速度的东东，执行后直接显示速度最快的服务器IP，是不是很爽？

## if exist

@echo off

if exist D:\BAT-Texy\test\\*.txt del D:\BAT-Texy\test\\*.txt

fdafd.bat 这个bat的文件名，实现循环执行，或监控动作

你只需要不关闭命令执行后的DOS窗口，不按CTRL+C强行终止命令，它就一直监视

## netstat 检查所有的网络端口状态

@echo off

netstat -a -n > a.txt

type a.txt | find "7626" && echo "Congratulations! You have infected GLACIER!"

del a.txt

pause & exit

这里利用了netstat命令，检查所有的网络端口状态，只需要你清楚常见木马所使用的端口，就能很easy的判断出来是否被人种了冰河。然这不是确定的，因为冰河默认的端口7626，完全可以被人修改。这里介绍的只是方法和思路。这里介绍的是方法和思路稍做改动，就变成可以检查其他木马的脚本了，再改动一下，加进去参数和端口及信息列表文件后，就变成自动检测所有木马的脚本了。呵呵，是不是很过瘾？脚本中还利用了组合命令&&和管道命令|，后面将详细介绍。

## ：： 注示

## cls 清屏

::clean screen

## 示例

@echo off

::close echo

cls

::clean screen

echo This programme is to make the MASM programme automate

::display info

echo Edit by CODERED

::display info

echo Mailto me : qqkiller\*\*\*@sina.com

::display info

if "%1"=="" goto usage

::if input without paramater goto usage

if "%1"=="/?" goto usage

::if paramater is "/?" goto usage

if "%1"=="help" goto usage

::if paramater is "help" goto usage

pause

::pause to see usage

masm %1.asm

::assemble the .asm code

if errorlevel 1 pause & edit %1.asm

::if error pause to see error msg and edit the code

link %1.obj & %1

::else link the .obj file and execute the .exe file

:usage

::set usage

echo Usage: This BAT file name [asm file name]

echo Default BAT file name is START.BAT

::display usage

批处理自动清除系统垃圾

@echo off

if exist c:\windows\temp\\*.\* del c:\windows\temp\\*.\*

if exist c:\windows\Tempor~1\\*.\* del c:\windows\Tempor~1\\*.\*

if exist c:\windows\History\\*.\* del c:\windows\History\\*.\*

if exist c:\windows\recent\\*.\* del c:\windows\recent\\*.\*

## Echo反馈

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

在实际应用中我们会把这条命令和重定向符号（也称为管道符号，一般用> >> ^）结合来实现输入一些命令到特定格式的文件中.

echo命令的另一种用法一：可以用它来显示信息！如例五中倒数第二行，Default BAT file name is START.BAT将在脚本执行后的窗口中显示，而echo命令本身不显示（为什么？？）。

echo命令的另一种用法二：可以直接编辑文本文件。例六：

echo nbtstat -A 192.168.0.1 > a.bat

echo nbtstat -A 192.168.0.2 >> a.bat

echo nbtstat -A 192.168.0.3 >> a.bat

把结果输入到指定的位置或文件中。

## @ 命令 - 不显示@以后的命令并执行

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。)

## autoset 当需要确认时，效果和/y是一样

@format X: /q/u/autoset

## :和goto

goto是个跳转命令，:是一个标签。当程序运行到goto时，将自动跳转到:定义的部分去执行了

if "%1"=="help" goto usage

pause

masm %1.asm

:usage

goto命令的另一种用法一：提前结束程序。在程序中间使用goto命令跳转到某一标签，而这一标签的内容却定义为退出。如：

……

goto end

……

:end

这里:end在脚本最后一行！其实这个例子很弱智，后面讲了if命令和组合命令你就知道了。

## Rem注释命令

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.

## Pause暂停命令

让当前程序进程暂停一下，并显示一行信息：请按任意键继续. . .。

在例五中这个命令运用了两次，第一次的作用是让使用者看清楚程序信息，第二个是显示错误的汇编代码信息（其实不是它想显示，而是masm程序在显示错误信息时被暂它停了，以便让你看清楚你的源代码错在哪里）。

## Call 命令

在批处理脚本中，call命令用来从一个批处理脚本中调用另一个批处理脚本。

start.bat：

……

CALL 10.BAT 0

……

10.bat：

……

ECHO %IPA%.%1 >HFIND.TMP

……

CALL ipc.bat IPCFind.txt

ipc.bat：

for /f "tokens=1,2,3 delims= " %%i in (%1) do call HACK.bat %%i %%j %%k

1、脚本调用可以灵活运用，循环运用、重复运用。

2、脚本调用可以使用参数！

在start.bat中，10.bat后面跟了参数0，在执行时的效果，其实就是把10.bat里的参数%1 用0代替。

CALL 10.BAT 0

ECHO %IPA%.%1 >HFIND.TMP -> ECHO %IPA%.0 >HFIND.TMP

在10.bat中，ipc.bat后面跟了参数ipcfind.txt（一个文件，也可以做参数），执行时的 效果，就是用ipc.bat中的每一行的三个变量，对应代换ipc.bat中的%%i、%%j和%%k。

## start 调用外部程序

调用外部程序，所有的DOS命令和命令行程序都可以由start命令来调用。

　入侵常用参数：

　　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 优先级类别开始应用程序

　　REALTIME 在 REALTIME 优先级类别开始应用程序

　　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。

## choice用户输入一个字符，从而运行不同的命令

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。

## If 命令

IF [NOT] string1==string2 do command

IF [NOT] EXIST filename do command

IF [NOT] ERRORLEVEL number do command

(1)、输入判断

if "%1"=="" goto usage 如果输入参数为

if not "%1"=="" goto usage 如果输入参数不为

(2)、存在判断

if exist C:\Progra~1\Tencent\AD\\*.gif del C:\Progra~1\Tencent\AD\\*.gif

(3)、结果判断

masm %1.asm

if errorlevel 1 pause & edit %1.asm

link %1.obj

先对源代码进行汇编，如果失败则暂停显示错误信息，并在按任意键后自动进入编辑界面；否则用link程序连接生成的obj文件。

先判断前一个命令执行后的返回码（也叫错误码，DOS程序在运行完后都有返回码），如果和定义的错误码符合（这里定义的错误码为1），则执行相应的操作（这里相应的操作为pause & edit %1.asm部分）。另外，和其他两种用法一样，这种用法也可以表示否定。用否定的形式仍表达上面三句的意思，代码变为：

if exist %1 (echo had!

) else (echo nothing!)

## errorlevel 执行命令返回码，表示是否出错

0 成功， 1 失败

## find搜索命令

用来在文件中搜索特定字符串，通常也作为条件判断的铺垫程序（我怎么突然想起了这四个字？）。这个命令单独使用的情况在批处理中是比较少见的，因为没什么实际意义。还是借例三来说明：

@echo off

netstat -a -n > a.txt

type a.txt | find "7626" && echo "Congratulations! You have infected GLACIER!"

del a.txt

pause & exit

先用netstat命令检查是否有冰河默认的端口7626在活动，并把结果保存到a.txt中。然后使用type命令列出a.txt中的内容，再在列出的内容中搜索字符串“7626” ，发现有的话则提示中了冰河，否则退出。看，find命令其实就这么简单，但有一点必须要注意到：如果不使用type命令列出a.txt中的内容，而是直接使用find命令在a.txt中找“7626”（find a.txt "7626" && echo "Congratulations! You have infected GLACIER!"），就必须得给出这个a.txt的绝对路径（我试过了，find并没有默认路径就是当前路径的功能，必须手动指定。也许是我错了，欢迎指正）。因为在find命令的帮助里有这么一句话：如果没有指定路径，find将搜索键入的或者由另一个命令产生的文字。这里的“另一个命令”自然就指的type命令了。

## type命令列出a.txt中的内容

列出文件内容

## 参数%命令

## | 通道命令

A | B , A的运行结果（输出）作为B的参数（输入）使用。

## || 通道命令

A||B,A运行 失败 则运行B

## & 通道命令

A&B,A运行完接着运行B，不论A是否运行成功

## && 通道命令

A&&B,A运行成功则运行B，否则不运行B