

HxC
2010

www.hackxcrack.es

Mrobles

BATCH

2

Bueno en este tutorial intentaremos entrar un poco más en profundo en batch y tocaremos

algunos códigos más útiles.



INDICE

INDICE	1
PROLOGO	1
1^a Fase: Básico-Medio	2
Start "Nuevo titulo" [OPCIONES] comando argumentos	2
IF (ERRORLEVEL y COMPARACION EN MINUSCULA)	3
& y	3
Msg	4
REDIRECCIONES ">", "<" y ">>"	4
Mode con y line	5
Del	5
Copy [opciones] [origen] [destino]	5
Move	6
REN	6
REM y ::	6
Variables de entorno	6
Comodines y mascaras	7
Shutdown	7
Tasklist	8
Taskkill	8
Setlocal y endlocal	8
CALL	8
Assoc	9
Attrib	9
At	9
Edit	10
Mem	10
Dir	10
Filtros	10
Type	10
2^a Fase: Medio-ALTO	11
Parametria (Parametro %0)	11
For	11
REG	12
ENcriptando en BATCH	14
ENABLEDELAYEDEXPANSION	17
FIND	17
Findstr	¡Error! Marcador no definido.

PROLOGO

Bueno en este tutorial intentaremos comprender un poco más la consola de comandos MS-DOS y profundizar en algunos códigos batch al igual que aprender otros para hacer un programa un poco más potente.

No será fácil, empezare con cositas que pueden parecer difícil pero llegado a el 2/3 de el tutorial explicare como poder hacer todo mas fácil aunque al final empezare con comandos muy útiles pero complejos que se suelen usar cuando se requiere de una gran complejidad y potencia para hacer programas

Bueno por ultimo solo recordarles que si ponemos **Help [comando]** o **[comando] /?** Todo ello sin los [] (ejemplo help cd o cd /?) nos dara información sobre el comando.

1ª Fase: Básico-Medio

Para comenzar profundizaremos en algunos códigos que Alan++ puso en el primer tutorial.

Start "Nuevo titulo" [OPCIONES] comando argumentos

Este comando tiene muchas mas posibilidades de las que se le suele dar
"Nuevo titulo" especifica el titulo de la nueva ventana que se generara.

Las opciones son las siguientes:

/D -ruta La dirección donde se ejecutara el programa
/B - El programa se inicia sin ventana
/I - Se inicia el programa original sin las modificaciones hechas en esa sesión.
/MIN - La nueva ventana se inicia minimizada
/MAX - La nueva ventana se inicia maximizada

(si sabéis ASM entenderéis mejor estos 2)

/SEPARATE - El programa se inicia en una zona de memoria separada
/SHARED - El programa se inicia en una zona de memoria compartida

Iniciar en prioridades:

Iniciar en prioridades:
/LOW - baja
/NORMAL - normal
/HIGH - alta
/REALTIME - tiempo real
/ABOVENORMAL - sobre lo normal
/BELOWNORMAL - debajo de lo normal
/WAIT - Inicia el programa, y espera hasta que termine de ejecutarse

Ahora os pondré un ejemplo de como abrir un programa sin que salga una ventana.

```
start /B /SEPARATE /HIGH /I /D C:\Windows\System32 nc -L -p 1337 |exit
```

Una vez ejecutado no veréis nada asi que no os extrañéis pero lo que hace es abrir netcat (si lo intentáis probar sin tenerlo instalado dara fallo) sin ventana y darle una prioridad alta en una

zona de memoria separada. Lioso ¿verdad? Jaja no os preocupéis es normal además que son comandos útiles pero solo cuando el nivel que se requiere es alto.

IF (ERRORLEVEL y COMPARACION EN MINUSCULA)

Bueno aquí solo aprenderemos por lo alto 2 variantes del IF común.

IF ERRORLEVEL 1 @echo ok ELSE @echo no

%ERRORLEVEL% es una Variable de Entorno que indica el nivel de error mostrado por un comando a su salida, no se extrañen si no entienden nada, yo tampoco xD , bueno se lo explico de otra manera:

Todo comando cuando es ejecutado produce un 1 o un 0 dependiendo de si se ejecuta bien o mal. Se usa normalmente con el IF para hacer que si un comando se ejecuta correctamente se realice una opción y si no es así otra.

En conclusión al poner este código devolverá un “ok” si se ejecuto bien y un “no” si lo hizo mal.

(El errorlevel puede dar hasta un 5)

IF /I "%~1"=="hola" echo hola

No se asusten xD, usamos %~1 para que aunque el usuario ponga comillas, no salga error wink y /I para hacer mas estricta la comparación por lo que este comando lo que haría es que solo si pone hola con minúsculas seria correcto.

& y |

Bueno Alan++ ya explico antes el comando & que servia para unir dos códigos en la misma línea por lo que yo les explicare para que sirve && y | .

&&

Este código hace lo mismo que && (unir 2 líneas de códigos) pero solo se ejecutara el segundo si el primero se ejecuta bien.

Ejemplo:

```
cd C:\Güindows&&echo bien&&pause
```

```
exit
```

Saldrá un bien solo si se está la carpeta Güindows y sino se saldrá directamente.

| (se escribe pulsando Alt GR + 1)

Este comando llamado PIPE o tubería es como & solo que en vez de primero una acción y luego otra, se ejecutan las dos a la vez.

Ejemplo:

Start notepad|start cmd

Se abrirá un blog de notas y una cmd a la vez, ya le daremos mas uso luego.

Msg

Ya dimos este code antes y lo que hacia era mandar un mensaje, pero solo vimos MSG * "mensaje" (las comillas no son necesarias)

La estructura es asi:

```
{nombreusuario | nombresesión | idsesión | @archivo | *(este para mandarlo a todos)}  
[/SERVER:servidor] [/TIME:segundos] [/V] [/W] [mensaje]
```

Lo dejo para que lo sepan porque suele ser útil por ejemplo para mandar mensaje solo a un compañero (se llamaría mrobles :P) de la red seria: MSG mrobles "Fiesta en mi casa"

REDIRECCIONES ">", "<" y ">>"

Ya vimos que se ponía pause>nul para que no apareciera el mensaje, ¿pero porque sucede eso?. Pues es simple solo redireccionamos el mensaje a la nada XD

DISPOSITIVO	SALIDA
NUL	Salida nula (a ninguna parte)
CON	Pantalla
PRN	Impresora por defecto
LPT1	1ª impresora en paralelo
COM1	1 puerto en serie
COM2	2 puerto en serie

El ">" hará que mandemos una orden a ese dispositivo ignorando lo que ya hay.

El ">>" hará lo mismo que el anterior pero guardará lo que había anteriormente y escribirá al final.

El "<" se usa para traer datos de otros dispositivos.

Ejemplo:

```
Echo ola>ejemplo.bat
```

```
Echo Adios>>ejemplo.bat
```

```
Pause>nul
```

Esto hará que haga un archivo llamado ejemplo.bat (si existe solo escribirá borrando lo de dentro) y escribirá ola, después adiós y por último se pausará pero no saldrá el mensaje de pulse cualquier tecla para continuar dado que lo redireccionamos a una salida nula.

Bueno ahora empezamos ya con lo nuevo

Mode con y line

Bueno ya aprendimos otros códigos para mejorar el aspecto como TITLE o COLOR hará aprenderemos a darle las dimensiones que nosotros queramos.

Es muy sencillo pondremos (cada uno que ponga lo que quiera en la parte roja):

MODE CON COLS=**41** LINES=**20**

Y nos cambiara el tamaño a lo que nosotros queramos

Del

Este puede ser uno de los comandos mas útiles que puede haber, sirve para borrar archivos (no carpetas) y tienes las siguientes opciones:

/P Pide confirmación.

/F Borra incluso los archivos de solo lectura (FORZAR BORRADO)

/S Borra también los subdirectorios.

/Q No pide confirmación en ningún caso (MODO SIENCIOSO)

Así, si quisieremos borrar todos los archivos de la carpeta sin permiso pondríamos:

del /S /Q *.*

Copy [opciones] [origen] [destino]

Su función es como su nombre indica copiar archivos

Opciones:

/A Indica un archivo de texto ASCII.

/B Indica un archivo binario.

/D Permite que el archivo destino se grabe descifrado

/N Al copiar usa el nombre corto en vez del nombre largo

/Z Copia archivos de red en modo reinicioable

/S Copia subdirectorios

/E Crea subdirectorios en el destino aunque los subdirectorios originales estén vacíos.

/V Verifica que los nuevos archivos se escriben correctamente.

/Y Suprime el mensaje para confirmar que desea sobrescribir un archivo de destino existente.

-Y El mensaje para confirmar que desea sobrescribir un archivo de destino existente.

(El modificador / Y puede estar preestablecido en la variable de entorno COPYCMD. Esto puede ser anulado con -Y en la línea de comandos.)

Ejemplos de lo que se puede hacer:

```
copy "C:\windows\archivo1.bat" "C:\archivo2.bat"
```

```
COPY /B "imagen.jpg" + "archivo.rar" imagenconrar.jpg
```

El primero solo copiara el archivo1 a la raíz y el segundo copiara pero a la vez unira dos archivos quedando una imagen con el código de un .rar en binario por lo que si le diéramos a abrir con WinRAR en vez del visor de imágenes se nos abriría lo que había dentro del .rar

Move

Se utiliza para mover archivos de un lugar a otro.

Ejemplo:

```
Move C:\carpeta\nombre.txt C:\carpeta2\nombre.txt
```

REN

Cambia el nombre de archivos (mas claro el agua)XD

Ejemplo:

```
REN archivo1.txt archivo2.txt
```

Cambiara el nombre del archivo1.txt por archivo2.txt

REM y ::

El comando REM(no confundir con el anterior ren) o :: se utiliza para dejar comentarios que no afectaran en nada para el código pero pueden ser útil si es muy largo y quieres dejar algunas guias para saber que es lo que hace cada parte.

Ejemplo:

```
@echo off  
  
REM ola  
  
REM ahora con puntos  
  
:: TAMPOCO SE VE  
  
Echo SHHHHHHHHHHHHH!!!! No as visto nada  
  
Pause>nul  
  
Exit
```

Solo se vera el SHHHHHHHHHHHHH!!! No as visto nada

Variables de entorno

Existen muchas variables predeterminadas ya por el navegado, se conocen como varialbes de entorno. Aqui les dejo algunas:

%RANDOM% Un numero aleatorio entre 0 y 32767
%DATE% La fecha
%TIME% La hora
%SYSTEMDRIVE% C:\
%HOMEDRIVE% C:\

%SYSTEMROOT% C:\Windows
%WINDIR% El directorio de Windows
%COMSPEC% C:\WINNT\system32\cmd.exe
%PROGRAMFILES% Archivos de programa
%TEMP% C:\DOCUME~1\Toni\CONFIG~1\Temp
%HOMEPATH% o %USERPROFILE% Carpeta del usuario (c:\documens and settings\"nombre")
%CD% Directorio actual
%COMPUTERNAME% Nombre del ordenador
Tambien estan estos que aunque no son variables son útiles aveces
VER Version que estas usando
VOL Volumen que estas usando

Comodines y mascaras

Existen 2 comodines * (equivale a mas de un caracter) y ? (equivalente a un solo caracter)

EJEMPLOS SACADOS DEL MANUAL DE HACKXCRACK DE NEDDIH

*.cfg seleccionar todos los archivos que tengan la extension cfg
a*.cfg seleccionar todos los archivos que comienzen por a y tengan la extension cfg
a*b.cfg seleccionar todos los archivos que comienzen por a, acaben por b y tengan la extension cfg
asa.cfg seleccionar todos los archivos que contengan 'asa' y tengan la extension cfg
c?asa.cfg seleccionar todos los archivos que contengan una C, luego un caracter cualquiera y luego 'asa'. Debe tener tambien la extension cfg
c?b*.* seleccionar todos los archivos que empiezen por c, tengan un caracter cualquiera, luego una b y cualquier extension
*.b?t seleccionar todos los archivos que tengan una extension que empieze por b, luego un caracter cualquiera y luego una t.

Shutdown

Este comando lo que hace es apagar el ordenador. ES el comando mas usado para empezar en mi opinion y si poneis en youtube como hacer un virus posiblemente lo que te aparezca es como hacer un acceso directo a este comando.

La verdad es un comando que no tiene gran complicación asi que explicare unas cosillas y si teneis curiosidad sobre como usarlo mas complejamente poned shutdown /? Como dije al principio.

Ejemplo:

```
shutdown -s -t 05 -c "hola hackxcrack"
```

La “-s” significa que se apagara ”-t 05” es el tiempo en segundo y -c “hola hackxcrack” es el mensaje que aparecerá. Si quisieramos que no apareciera pantalla con poner shutdown -s -t 0 bastaria y también podríamos hacer que se reiniciara en vez de apagarse cambiando la -s por -r (shutdown -r -t 0).

PD: Para anular se pone shutdown -a y para forzar apagado podemos añadir un -f

Tasklist

Muestra el nombre, PID, nombre de sesiones y el uso de memoria de los procesos que están actualmente en el ordenador.

Taskkill

Para matarprocesos se utiliza este comando, es muy útil dado que si un archivo esta en uso no se podrá borrar por lo que primero lo matamos y luego lo borramos (que gánster suena esto)

La sintaxis es la siguiente TASKKILL [/S sistema] [/U usuario [/P contraseña]]
{ [/FI filtro] [/PID IdProceso | /IM NombrelImagen] } [/F] [/T]

Normalmente con poner por ejemplo taskkill nombre.exe nos bastara pero no esta de mal saber que hay mas opciones

Setlocal y endlocal

Setlocal y endlocal indican que un espacio local en el que trabajaran las variables (setlocal indica el inicio de ese espacio y endlocal el final). Con un ejemplo lo entenderán mejor.

```
@echo off  
  
Set var=423423  
  
Setlocal  
  
Set var=13412  
  
Endlocal  
  
Echo %var%  
  
Pause>nul
```

Hemos cambiado la variable al final pero como solo afecta al espacio comprendido entre setlocal y endlocal y el echo %var% esta fuera pondrá la variable que había e principio.

CALL

Lo que hace es una llamada a una de las etiquetas o a un archivo.

Ejemplo:

```
@echo off  
  
Echo ola  
  
Call :salto  
  
Pause>nul  
  
exit  
  
:salto  
  
Cls
```

Echo HackxCrack

En la pantalla solo se mostrara hackxcrack. Es como si el código fuera este:

```
@echo off
```

```
Echo ola
```

```
ClS
```

```
Echo HackxCrack
```

```
Pause>nul
```

```
Exit
```

Assoc

Con este comando podemos crear nuestras propias extensiones o modificar las ya existentes.

La sintaxis es muy simple, assoc .[EXTENSION]=[PROGRAMA PARA ASOCIAR].

Ejemplo:

```
assoc .txt=batfile
```

```
assoc .jpg=notepad.exe
```

Ahora los txt se abrirán con la cmd (como veis no tiene por que ser el nombre del programa sino que tambien hay ya algunos nombres predeterminados como el de la cmd aunque si pusieras cmd.exe funcionaria igual) y las imágenes en jpg con el blog de notas.

Attrib

Sirve para cambiar las propiedades de los archivos.

H - Atributo oculto. Se activa con +h y desactiva con -h

R – Atributo de lectura. Solo lectura con +r, normal con -r

S – Atributo de sistema. Se activa con +s y desactiva con -s

A – Atributo modificado. Para copias de seguridad incrementales. +a indica modificado y -a indican no modificado.

ATTRIB archivo.txt Muestra los atributos del archivo

Ejemplo:

```
attrib +R +A +S +H *.*
```

```
attrib -R -A -S -H *.*
```

Con el primero pondriamos todos los atributos ocultando todos los archivos y con el Segundo los quitariamos pudiendo verlos ya.

At

El comando AT permite automatizar la ejecución de un comando para una hora y/o

fecha determinada. Por ejemplo que el dia 1/1/11 a las 11:11 borre todo. O que a las 5 de todos los dias se apague el pc. O tambien que se ejecute algo todos los días 30. Aquí les mostrare unos ejemplos y veran los facil que es, solo hay que saber que every indica que es siempre, next que es el próximo y aunque no lo use para que lo sepan, interactive para que se haga aunque el usuario este haciendo cosas.

```
at 5:00 shutdown -s -f -t 0  
at /next:30 5:00 shutdown -s -f -t 0  
at /every:4 5:00 shutdown -s -f -t 01
```

El primero indica que a las 5 se apagara el ordenador, el Segundo que el proximo dia 30 se apagara y el tercero que todos los dias 4 se apagara.

PD: Para borrar una tarea solo hay que poner AT /delete

Edit

Este comando es un simple editor como el blog de notas pero en MS-Dos, solo hay que poner edit en la consola y aparecerá solo.

Mem

Otro comando que no es de mucha utilidad pero que de vez en cuando se usa. Lo que hace es mostrar la memoria del ordenador (es uno de los pocos comandos que no esta ni traducido).

/P Muestra memoria usada por programas.
/D Muestra memoria usada por programas y controladores.
/C Muestra memoria por tamaño

Dir

Este comando muestra los archivos y subdirectorios de una ubicación. Es un comando bastante usado e interesante.

Filtros

Los filtros son órdenes que sirven para formatear la salida del comando de acuerdo a nuestros intereses. (Antes de poner el filtro hay que poner una tubería o pipe "|")

El filtro SORT sirve para ordenar la salida.

DIR | SORT

El filtro MORE sirve para pausar la salida cada pantalla para que pueda ver todos los datos de salida con tranquilidad. Pasa de pantalla a pantalla al pulsar una tecla.

DIR | MORE

Type

Sirve para imprimir en pantalla el contenido de un archivo.

Ejemplo:

Tenemos un archivo llamado "lol.txt" en el que dentro hemos escrito "Hola mundo estoy aquí jugando con la consola para probar los comandos"

Bien, pues si entramos en la cmd y nos vamos hasta la ruta donde este el archivo y ponemos type lol.txt nos aparecerá en la consola “Hola mundo estoy aquí jugando con la consola para probar los comandos”

2ª Fase: Medio-ALTO

Ahora empieza lo complejo aunque también habrá cosas sencillas al igual que en la anterior fase había complejas.

Parametria (Parametro %0)

¿Qué pasa si no sabemos en la ruta en la que va a estar nuestro archivo bat? Pues fácil para eso tenemos el paramtro especial %0 en el que esta guardada la ruta del archivo actual. Veamos un ejemplo:

```
copy %0 C:\Windows\System32
```

Con este simple code ya tendremos nuestro archivo en Syste32, muy útil para hacer wares.

```
Copy %0 C:\\"Documents and Settings"\\"All Users"\*  
Inicio\\"Programas"\\"Inicio"\\"mivirus.bat"
```

Esta ultima es la carpeta en la que están alojados los archivos que se inician automáticamente en cuanto encendemos el ordenador y es muy útil, asi que se la pongan.

For

Coged provisiones que este es largo dado que es de los mas complejos.

La forma general es FOR %%variable IN (texto1.txt texto2.txt texto3.txt) do [COMANDOS]

Por ejemplo:

```
FOR %%x IN (texto1.txt texto2.txt texto3.txt) do delete %%x
```

Borrara el texto1,2 y 3

Ahora iremos con FOR avanzado

Esta vez el comando For viene con el modificador /F, esta es la sintaxis

```
For /F opciones %%a in (archivos) do comandos
```

Expliquemos:

/F: Modificador

opciones: "reglas" Despues comprendereis

%%a: variable

(archivos): archivos con los que se trabajaran

comandos: comandos que se ejecutaran

Os doy las opciones:

EOL=x Indica que se realice la acción en todos los archivos menos los que empizan por esa letra

SKIP=x Indica la linea por la que se empezara (Ejemplo: Skip=3 las 3 primeras lineas no recibirán la acción)

DELIMS=xxx Indica en que simbolo o letra acaba la acción del for

TOKENS=x,y,z Indica las vueltas que seran validas.

Veamos un ejemplo:

Creamos un txt (el de el ejemplo es archivo.txt asi que si quereis probar debéis guardar con ese nombre) que contenga:

```
HackxCrack  
Hola - mundo  
Adios-mundo  
Hasta luego-gebte
```

Y ahora un bat que contenga:

```
@echo off  
For /F "DELIMS== SKIP=1" %%a in (archivo.txt) do echo %%a
```

```
Pause>nul
```

¿Esto que hará? Nos mostrara las lineas del archivo excepto la primera y acabara de procesar cada linea en - . Por tanto nos mostrara: Hola Adios Hasta Luego.

¿No es tan dificil no?

REG

Este es sin duda un comando que no puede faltar en ningún malware y en algunos software tampoco. Reg lo que nos permite es modificar el registro de Windows automáticamente al igual que cuando ponemos REGEDIT y lo hacemos a mano.

Sintaxis Reg [operacion] [parametros]

Operadores y parametros:

ADD - Para agregar una clave al registro

```
ADD Clave [/v nvalor | /ve] [/t tipo] [/s separador] [/d datos] [/f]
```

Clave = \\equipo\ClaveRaiz\SubClave ClaveRaiz=HKLM, HKCU, HKCR, HKU, HKCC
/v = Nombre del valor a agregar
/ve = agrega el valor vacío (sin nombre)
/t = 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)
/s = carácter. (si se omite se asume \0)
/d = datos que se agregan al valor a insertar
/f = fuerza la sobreescritura sin avisar

DELETE – Borra un registro.

DELETE clave [/v nvalor | ve] [/va] [/f]

Clave =ClaveRaiz\SubClave ClaveRaiz=HKLM, HKCU, HKCR, HKU, HKCC
nvalor: nombre de valor a borrar. Si se omite se borrarán todas las subclaves y valores
/ve elimina el valor de un nombre de valor vacío
/va elimina todos los valores en la clave actual
/f fuerza la eliminación sin avisar

QUERY - Para consultar una clave

REG QUERY clave [/v nvalor | /ve] [/s]

Clave = \equipo\clave Si se omite equipo se usa el equipo local.

En equipos remotos solo disponibles HKLM y HKU

/v = consulta para una clave de registro específica

/ve = consulta el valor predeterminado

`/s` = consultar todos los valores/subclaves

EXPORT - Para exportar claves y valores.

EXPORT clave archivo

Clave =ClaveRaiz\SubClave

ClaveRaiz=HKLM, HKCU, HKCR, HKU, HKCC

Archivo: archivo donde exportar

IMPORT - Importa una clave exportada anteriormente con EXPORT.

IMPORT archivi

El archivo debe haber sido creado con export

archivo = archivo.reg (solo equipo local)

COMPARE - Compara una clave con otra.

COMPARE clave1 clave2 [/y valor | /ye] [salida] [/s]

Clave = \\equipo\ClaveRaiz\SubClave ClaveRaiz=HKLM, HKCU, HKCR, HKU, HKCC

Valor = nombre del valor para comparar en la clave seleccionada (si se omite se

comparan todos)

/ve = Comparar el nombre de valor vacío

`/s` = comparar todas las subclaves y valores

salida [/oa | od | os | on] *oa=devuelve todas las coincidencias y diferencias,
od=solo diferencias, os=solo coincidencias, on=nada

SAVE SAVE – Guarda en un archivo del árbol de redistro.

SAVE **SAVE** clave archivo

Clave=ClaveRaiz\SubClave ClaveRaiz=HKLM HKCU HKCR HKU HKCC

SubClave=nombre completo de la subclave

Archivo = Nombre del archivo para guardar.

LOAD – Carga un archivo en el árbol

LOAD clave archivo

Clave =ClaveRaiz\SubClave

ClaveRaiz=HKLM, HKCU (solo para equipo local)

Archivo: creado con reg save

UNLOAD – Descarga un árbol del registro.

UNLOAD clave

Clave =ClaveRaiz\SubClave

ClaveRaiz=HKLM, HKCU (solo para equipo local)

COPY – Copia una clave en el registro

COPY clave1 clave2 [/s] [/f]

Clave =ClaveRaiz\SubClave

ClaveRaiz=HKLM, HKCU, HKCR, HKU, HKCC

/s = copia todas las subclaves y valores

/f = fuerza la copia sin avisar

RESTORE – Restaura una clave

RESTORE clave archivo

Clave =ClaveRaiz\SubClave

ClaveRaiz=HKLM, HKCU, HKCR, HKU, HKCC

Archivo: Nombre del archivo a restaurar. Este archivo debe haber sido creado con reg save

EJEMPLOS:

Reg Add HKLM\Software\Microsoft\Windows\Currentversion\Run /v programa.bat /t reg_sz /d C:\windows\system32\programa.bat /f

Agrega el archivo “programa.bat” a la entrada donde estan todos los programas q se inician con Windows que esta en system32 y fuerza la escritura.(Es lo que se utiliza en Malwares)

Reg Delete HKLM\Software\Microsoft\Windows\Currentversion\Run /v programa.bat

Borra lo que pusimos con el Reg Add anterior

Reg Query HKLM\Software\Microsoft\Rest /v Version

Muestra la version del registro

Reg Export HKLM\Software\Mico\MiAP CopiaAP.reg

Exporta CopiaAP.reg

Reg Import CopiaAP.reg

Importa CopiaAP

Reg Compare HKLM\Software\Mico\MiAP HKLM\Mico\MiAP2

Compara los valores de MiAP con MiAP2

Reg Load HKLM\Software\Mico\MiAP SubAP.hiv

Guarda el subarbol MiAP en el archive SubAP.hiv en la carpeta actual

Reg Unload /HKLM\Software\Mico \MiAP

Descarga el arbol MiAP del registro

Reg Copy HKLM\Software\Mico\MiAP HKLM\Software\Mico2

Copia todas las subclaves y valores de MiAP a Mico2

Reg Restore HKLM\Software\Mico\MiAP\Datos MiAP2.hiv

Restaura el archive MiAP2.hiv sobreescriviendo la clave “datos”

Reg Save HKLM\Software\Mico\MiAP MiAP2.hiv

ENcriptando en batch

Esto en realidad sirve para hacer wargames o para que no venga el típico lammer y se limite a copiar y pegar cambiando el titulo por el suyo y dándoselas de ser el creador.

Se puede arreglar copilando a .exe ya que se encripta mas potenteamente pero este método queda mas elegante y además la gente que si sepa de batch podrá comprender tus códigos mientras que el RIPER (El que se dedica a copiar y poner su nombre) no podra.

Empezamos con algo simple como esto

```
@echo off  
set uno=echo  
set dos=Hola  
%uno% %dos%  
Pause>nul
```

Se vera solo Hola dado que si sustituimos %uno% %dos%, quedaría un simple echo Hola.

Facil, ¿no?

Ahora agamos un programa de contraseñas o wargame.

```
@echo off  
set a=Crack  
set b=Hack  
set c=x  
Echo Contraseña:  
set/p pass=%>%  
REM EL %>% se pone porque > es un carácter especial  
if "%pass%"=="%b%%c%%a%" (echo.Bien) else (echo.Mal)  
REM "%pass%" la variable se mete entre comillas para evitar errores al  
REM intentar una contraseña entre comillas  
pause>nul  
exit
```

Si ponemos de contraseña HackxCrack nos dira bien sino pues mal.

Ahora lo complicaremos algo mas, haremos extracción de caracteres.

Extrac... ¿Qué?

Extraccion de caracteres, es muy fácil solo hay que poner

%variable:~PARTEMALA,partebuena%

Ahora te lo explico para que entiendas mejor

% = Indican que es una variable

variable = El nombre que tiene la variable que cogeremos

: = Indican que vamos a trabajar con la variable

~ = Indican que vamos a extraer caracteres

PARTEMALA= Los caracteres que no queremos (ejemplo: si es 2, no cogeríamos los 2 primeros)

partebuena= Los caracteres que si queremos (ejemplo: si es 3, cogemos los 3 siguientes

después de la parte que no queriamos)

Ejemplo:

```
@echo off  
set ABC=hasdHACKXCRACKasduasg  
set DEF=asHACKssXsasCRACKasdasd  
echo %ABC:~4,10%  
echo %DEF:~2,4%%DEF:~8,1%%DEF:~12,5%  
pause>nul  
exit
```

Nos mostrara 2 veces HackxCrack (intente evitar ponerle numeros a las variables como nombre porque puede confundirse con códigos de parametria)

Bueno ¿Lo habéis pillado? Pues ahora varias derivantes de esa estraccion.

-Si solo ponemos un numero al final se cogera como que la parte que no debe coger es el numero y todo lo demás será mostrado.

-Si solo hay un número y el es negativo empezara a contar de atrás a adelante eligiendo tantos caracteres como el valor del numero.

-Si hay 2 pero uno es negativo se contara de atrás a adelante y el resto normal.

Es algo lioso si no se ve con nuestras propias manos XD

Ejemplos

```
@echo off  
Set ABC=1234567890  
echo %ABC:~3%  
echo %ABC:~-3%  
echo %ABC:~-5,3%  
echo %ABC:~-5,-3%  
echo %ABC:~-6,-3%  
Pause>nul  
Nos devolverá 4567890      890      678      67      567
```

Ahora remplazaremos caracteres, la sintaxis es %variable:caracter=cambiado%

Carácter= El carácter que queremos remplazar

Cambiado= El carácter por el que lo queremos remplazar

Ejemplo:

```
@echo off  
set var=Mrobles  
set var2=%var:o=0%  
set var2=%var2:l=1%  
echo. %var% %var2%  
pause>nul  
exit
```

Mostrará: Mrobles Mr0b1es

Bueno ahora pongamos en practica lo de antes y hagamos una aplicación para crackme/password/wargame.

```
@echo off  
echo Contraseña:  
set/p Psw=%>%  
call :cmb  
if %psw%==Mr0b1es (echo.bien) else (echo.mal)  
REM Los caracteres especiales como % hay que rodearlos de con % para que se vean  
pause>nul  
exit  
:cmb  
set psw=%psw:l=1%  
set psw=%psw:o=0%  
goto:eof  
REM goto :eof vuelve a la función anterior
```

Si introducimos Mrobles nos dara bien sino pues mal XD

ENABLEDELAYEDEXPANSION

Este código se usa para expandir una variable (poder meter una variable dentro de otra) para activarlo hay que poner setlocal ENABLEDELAYEDEXPANSION y solo funciona dentro de el espacio local comprendido entre setlocal y endlocal

FIND

Este comando busca cierta cadena en un archivo (se puede usar como filtro)

Modificadores:

/v Muestra todo menos lo que contenga esa palabra.
/c
/n
/i
/offline

Ejemplo:

```
@echo off
title MROBLES GOSHT
MSG * "Gracias por entrar en la comunidad XD"
:ini
tasklist | find /i "www.hackxcrack.es" >nul && (goto:ini) || (start
www.hackxcrack.es&goto:ini)
```

CONCEMPTOS QUE FALTAN Y PROXIMAMENTE PUBLICARE:

findstr

net

ping

tracert

telnet

choice

paths

prompt

Propagacion USB y P2P

ftp

netstat

nbtstat

Nslookup

shift

OPERADORES EXTRAÑOS

POPD y PUSHD

grafismo

