Hola que tal, bueno como les prometi en algunos post ,y si nadie del vip se enoja les paso a contar
como es este metodo que se me ocurrio que viene hechando abajo a las heuristicas del nod y el Kav.
aunque todavia falta mucho por descubrir y la idea mia de hacerlo publico es para que alguien se anime
y empieze a investigar y jugar con cosas mas "avanzadas" , y viendo que no todo esta inventado, y debe haber mil cosas mas todavia
tan simples y que no vemos.

bueno empezemos:

en que se basa?

las heuristicas de los antivirus , saltan como loco cuando ven un archivo consola , que llama a procesos como Writeprocessmemory,
de forma automatica .

entonces?

no llamemos a nadie de afuera ,asi no alertamos a el antivirus :) , incluamos el codigo q busca afuera , en nuestro propio executable
al fin y al cabo son todas instrucciones no? , unas las busca afuera otras adentro, bueno hagamos q todas las haga adentro

vamos al olly , y ponemos **search , intermodular calls**, en este caso vamos a hacerla con una simple **unmaViewofsection**

nos ponemos arriba la funcion y apretamos enter, en este lugar esta la llamada a la funcion(lo q despues vamos a reemplazar)
apretamos enter 2 veces mas

```
00408916  . 50              PUSH EAX
00408917  . 8B45 C8         MOV EAX,DWORD PTR SS:[EBP-38]
0040891A  . 50              PUSH EAX
0040891B  . E8 E4FEFFFF     CALL <JMP.&ntdll.ZwUnmapViewOfSection>
00408920  . 85C0            TEST EAX,EAX
00408922  .v0F8C 1D010000   JL 1.00408A45
00408928  . 837D FC 00      CMP DWORD PTR SS:[EBP-4],0
0040892C  .v0F84 13010000   JE 1.00408A45
```

hasta que nos aparesca la función(aca ya estamos en la dll ,afuera del exe)

```
7C91DEEC  C2 1000           RETN 10
7C91DEEF  90                NOP
7C91DEF0  B8 0B010000       MOV EAX,10B
7C91DEF5  BA 0003FE7F       MOV EDX,7FFE0300
7C91DEFA  FF12              CALL DWORD PTR DS:[EDX]
7C91DEFC  C2 0800           RETN 8
7C91DEFF  90                NOP
7C91DF00  B8 0C010000       MOV EAX,10C
7C91DF05  BA 0003FE7F       MOV EDX,7FFE0300
7C91DF0A  FF12              CALL DWORD PTR DS:[EDX]
7C91DF0C  C2 0800           RETN 8
```

copiamos eso en una seccion nueva
y la agregamos al ejecutable , acordemosno de la posicion que empieza la seccion :)

volvemos al programa principal gralmente **goto 401000** ,
de nuevo **search - intermodular calls** y **enter** en la funcion
hacemos doble click en la llamada y ponemos
CALL "y aca la direccion de la nueva seccion"
(hacerlo con cada llamada)
guardamos y si todo salio bien :) , una dependencia externa menos.

Nota: con funciones mas complicadas, tenemos q arreglar todos los call de la seccion q agregamos

IMPORTANTE: este es un ejemplo a modo teorico : esta funcion si bien anda en sp2 y sp3 no puedo decir lo mismo
del vista o otras versiones de windows, tengan en cuenta que le estamos pasando una direccion a EDX que es propia
de la dll , y las dll cambian entre los windows.
que seria lo ideal? meter todo el codigo en la nueva seccion con todas sus llamadas y dependencias. si es un laburo largo
y puede llevar de varias horas a 1 mes , dependiendo la función.

pero no se desanimen hay muchas funciones simples , como getprocessid , que son solo funciones q mueven datos
y hasta sacan algunas heuristicas ;)
es cuestion de probar. si funciona el metodo es **indetectable** para todos las heuristicas, lo eh comprobado

espero que les sea util este metodo, y no dejen que los AV , lo averiguen!

EJEMPLO:

en este ejemplo le eh aplicado el metodo a la funcion SetHandleCount, los archivos estan mas abajo para descargar.

ARCHIVO ORIGINAL
Código:

```
00402358  |. FF15 30504000  CALL DWORD PTR DS:[<&KERNEL32.SetHandleC>;
\SetHandleCount

7C80CD27 > 8BFF            MOV EDI,EDI
7C80CD29   55              PUSH EBP
7C80CD2A   8BEC            MOV EBP,ESP
7C80CD2C   8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
7C80CD2F   5D              POP EBP
7C80CD30   C2 0400         RETN 4
```

ARCHIVO CON EL METODO
Código:

```
00402358     FF15 0E904000  CALL DWORD PTR DS:[40900E]              ;
Copia_de.00409000
.

00409000   8BFF            MOV EDI,EDI
00409002   55              PUSH EBP
00409003   8BEC            MOV EBP,ESP
00409005   8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
00409008   5D              POP EBP
00409009   C2 0400         RETN 4
0040900C   0000            ADD BYTE PTR DS:[EAX],AL
0040900E   0090 40000000    ADD BYTE PTR DS:[EAX+40],DL    ; PUNTERO A
409000  (hay q ponerlo en little endian   00904000 )
```

fijense como desaparece la funcion de todos lados ,del import y hasta el olly ni se da cuenta que se usa :)
lo mismo pasa con los AV.

en el 409000 esta la nueva seccion llamada .leo q contiene el codigo que copie desde el kernell32, osea la funcion sethandlecount
yo lo hice con puntero , asi es mas facil modificar el call original. y se simula lo mayor posible el programa.

Archivos Ejemplos:

http://www.sendspace.com/file/3n0l5g
y a mirar con los debugger :D.

si alguien puede verificar la compatibilidad con Vista , seria muy bueno.