Cracking Commands used in Softice 3.2

Written by : +Vip-Vop

Published by : Splatter Industries,Inc.

This tutorial covers some of the more important Softice commands for cracking. Although they are good commands to get familiar with don't limit yourself to just this lesson. Softice has a wide variety of commands and uses that should also be explored.

BPX

"bpx" means break point on execution. You can use the address of a line of code(ie: bpx 0041a536) or a call from a dll that you have exported in your winice.dat(ie: bpx messageboxa). Here are some common bpxs:

BPX's for time/date protections:

:bpx settimer

when settimer is called, if you scroll up a couple of lines of code, past a couple of pushes, one push pushes the length of the timer in milliseconds.

:bpx getsystemtime

gets the system time.

:bpx getlocaltime

gets the system time.

BPX's for breaking on nag boxes:

:bpx messagebox

pops up a messagebox

:bpx showwindow

if the nag screen isnt a msg box

BPX's when entering reg info:

:bpx getdlgitem

16-bit. sometimes used. gets value of text box

:bpx getdlgitemtexta

32-bit. used often. gets value of text box

:bpx getwindowtext(a)

16(32)-bit. used sometimes. gets value of window or text box. best to use others first.

:bpx getwindowtextlength(a)

16(32)-bit. gets length of string in box. might be used before getting the text, or after.

:bmsg param1 wm_command

param1 is the hwnd of the button. This is useful to see what happens right after you press OK.
To find the hwnd first type :task and find the name of the program you want.
Then do a :hwnd (name of task) and it will list the hwnd of all items on that task.
keep on guessing until you get the right button. This method is a pain in the ass
I don't recommend it.

BPX's for when a program reads from an .ini/the registry

:bpx regqueryvalueexa

gets the value of a registry entry

:bpx getprivateprofilestring

gets a line from an .ini

:bpx regopenvalueexa

I think that is the call. opens a reg entry

BPM

Breakpoint on Memory - if you have the read flag set, it breaks when that memory location is read, if you have the write flag set, it breaks when that location is written to. A bpm looks like this: bpm 00419425 RW
That would break whenever the memory at location 00419425 is read or written to. This is a useful breakpoint if you already know what the registration variable is (usually a location that needs to be 1 or 0), but you don't know where it's set. Just bpm on the location with the write flag (W) and soft ice will stop whenever it is set.

BL

Breakpoint list. Lists all the current breakpoints and their breakpoint number. The breakpoint number is used for clearing them, disabling them, and enabling them. If there is a "*" next to a breakpoint it means it's currently disabled.

BC

Breakpoint clear. Clears a breakpoint from memory. You can type "bc *" to clear all breakpoints, or "bc (breakpoint number)" to clear just a certain one.

BD,BE

Breakpoint disable, breakpoint enable. If you don't want to clear a breakpoint, but you don't want it to break right now, disable it. To find the breakpoint number use BL. BD and BE can also use * to disable/enable all.

D

D displays whatever you put after it. "d eax" would display the value of eax. "d 00419533" would display the value of memory location 00419533.

R

R is the register command. If you want to modify a register's value, say eax, you would type "r eax" then enter in the new data. If you want to change a flag, like the zero flag for a jnz/jz/jne/je type "r fl z". fl means flag, and the z stands for the zero register. The other flags aren't often used.

A

Assemble. Lets you change the program instructions at whatever address you type after the a.

F8

By default F8 is the "t", trace, command. F8 will step line by line through the code, and will trace into calls instead of just executing the call like F10 does. F8 isn't used as much as F10 becuase most of the time you don't want to trace into every call. But if you do want to trace into a call, you use F8.

F10

By default F10 is the "p", proceed, command. Press F10 to step line by line through code, though F10 goes over calls, instead of tracing into them like F8 does. This is the key you will be pressing most of the time.

F12

By defualt F12 is the "p ret", proceed to the next return, command. F12 will get you out of the call you are in and into whoever called it. Whenever you set a bpx on a windows system function, like messageboxa, you will press F12 right away to get back to your programs code and out of the windows system code.

INIT line for win-ice.dat

Here is a good INIT line for your win-ice.dat in your soft-ice directory:

```
INIT="lines 60;color f a 4f 1f e;wd 22;wc 22;wr;code on;faults off;x;"
```

You also need to go down to your exports section of win-ice.dat, and removing the ";" in front of the following dlls if you haven't already done so:

```
EXP=c:\windows\system\kernel32.dll
EXP=c:\windows\system\user32.dll
EXP=c:\windows\system\gdi32.dll
If you are cracking Visual Basic programs use these dlls too:
EXP=c:\windows\system\msvbvm50.dll
EXP=c:\windows\system\vb40032.dll
```

For more tutorials on cracking and reverse engineering please visit :
Splatter Industries,Inc. http://www.splatter.net
Or write to +VipVop c/o vipvop@hotmail.com

Cracking Commands used in Softice 3.2
Was written by : +Vip-Vop
And published by : Splatter Industries,Inc.
1998 All rights reserved.

The End