A few of my favorite things...debug commands

SpatDSG 7 May 2007 10:04 AM

2

Well. not really, I mean - not in the big picture right? Else that would be a sad existence indeed... talk about whacked priorities. I thought I would post a few debugger commands I like to use… some new to me, some oldies.

But, there are times I suppose when these really are my favorite things at that moment, when I know it will save me hours of otherwise useless work. When I used to build furniture (loved that job back in college) , it was an amazing thing when you used the right tool for the right job.. (chisel instead of screwdriver)
First of all , the ever useful FOR command…

Find all modules loaded in all processes in dump:

```
!for_each_process ".process @#Process;!peb"
```

Get all stacks for all processes

```
!for_each_process ".process /p /r @#Process;!process @#Process"
```

Check integrity of a binary (useful for those pesky corrupted binaries or hax0red even)

```
!for_each_module !chkimg @#ModuleName
```

Use the .shell command.. in this case to find a specific string in data

```
kd> .shell -i - -ci "!thread" findstr -c:"nt!"
Start Address nt!ExpWorkerThread (0x804e4196)
f9015d20 8067e3ac 00000007 805615c0 8056167c nt!RtlpBreakWithStatusInstruction (FPO:
[1,0,0])
f9015d74 804e426b 00000000 00000000 82bc4030 nt!ExpDebuggerWorker+0x91 (FPO: [Non-Fpo])
f9015dac 8057d0f1 00000000 00000000 00000000 nt!ExpWorkerThread+0x100 (FPO: [Non-Fpo])
f9015ddc 804f827a 804e4196 00000001 00000000 nt!PspSystemThreadStartup+0x34 (FPO: [Non-Fpo])
00000000 00000000 00000000 00000000 00000000 nt!KiThreadStartup+0x16
.shell: Process exited
```

Get stacks from all processes where win32k is listed:

```
kd> !stacks 2 win32k
Proc.Thread  .Thread  Ticks   ThreadState Blocker

Max cache size is       : 1048576 bytes (0x400 KB)
Total memory in cache   : 0 bytes (0 KB)
Number of regions cached: 0
0 full reads broken into 0 partial reads
    counts: 0 cached/0 uncached, 0.00% cached
    bytes : 0 cached/0 uncached, 0.00% cached
```

```
** User virtual addresses are translated to physical addresses before access
** Prototype PTEs are implicitly decoded

                            [82bc77c0 System]
*** ERROR: Module load completed but symbols could not be loaded for ino_fltr.sys
*** ERROR: Module load completed but symbols could not be loaded for userdump.sys

                            [82a8c380 smss.exe]

                            [829d5558 csrss.exe]
 22c.000234  82a32da8 0000000 Blocked    nt!KiSwapContext+0x2e
                                        nt!KiSwapThread+0x46
                                        nt!KeWaitForSingleObject+0x1c2
                                        Ntfs!NtfsWaitSync+0x1c
                                        Ntfs!NtfsNonCachedIo+0x30e
                                        Ntfs!NtfsCommonRead+0xbdd
                                        Ntfs!NtfsFsdRead+0x22d
                                        nt!IopfCallDriver+0x31
                                        sr!SrPassThrough+0x31
                                        nt!IopfCallDriver+0x31
                                        ino_fltr+0x7544
                                        nt!IoPageRead+0x1b
                                        nt!MiDispatchFault+0x274
                                        nt!MmAccessFault+0x5bc
                                        nt!KiTrap0E+0xcc
                                        win32k!bDynamicModeChange
                                        win32k!DrvChangeDisplaySettings+0x4de
                                        win32k!xxxUserChangeDisplaySettings+0x141
                                        win32k!RemoteSetDisconectDisplayMode+0x28
                                        win32k!xxxRemoteDisconnect+0x188
                                        win32k!NtUserCallNoParam+0x1b
                                        nt!KiFastCallEntry+0xf8
                                        ntdll!KiFastSystemCallRet
```

If you have private symbols you can see your own data type information, or the type info is in NT and the target is XP or greater..

```
kd> dt NT!*PROCESS*
          NT!_KPROCESSOR_STATE
          NT!_PROCESSOR_POWER_STATE
          NT!_EPROCESS
          NT!_KPROCESS
          NT!_EPROCESS_QUOTA_BLOCK
          NT!_SE_AUDIT_PROCESS_CREATION_INFO
          NT!_EPROCESS
          NT!_EPROCESS_QUOTA_ENTRY
          NT!_EPROCESS_QUOTA_BLOCK
          NT!_RTL_USER_PROCESS_PARAMETERS
          NT!_EPROCESS_QUOTA_ENTRY
    NT!PROCESSOR_IDLE_TIMES
```

In usermode – the "TC" command..
In this case I am in notepad.exe and want to "fast forward" my debugging to the next call in DialogBox2(), step over it and see the result then move on or re-examine it.

```
0:000> KL 3
ChildEBP RetAddr

0015f9b8 77491460 USER32!DialogBox2
0015f9e0 774914a2 USER32!InternalDialogBox+0xd0
0015fa00 774b12de USER32!DialogBoxIndirectParamAorW+0x37

0:000> r
eax=00141904 ebx=00000000 ecx=00141904 edx=008c0570 esi=00000001 edi=000c18fe
eip=77491244 esp=0015f9bc ebp=0015f9e0 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000202
USER32!DialogBox2:
77491244 8bff            mov     edi,edi

0:000> tc
eax=00141904 ebx=00000000 ecx=00141904 edx=008c0570 esi=00000001 edi=000c18fe
eip=7749125e esp=0015f990 ebp=0015f9b8 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000202
USER32!DialogBox2+0x16:
7749125e e8ed080100      call    USER32!ValidateHwnd (774a1b50)

0:000> p
eax=009bde28 ebx=00000000 ecx=00062c30 edx=008c0501 esi=00000001 edi=000c18fe
eip=77491263 esp=0015f990 ebp=0015f9b8 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000206
USER32!DialogBox2+0x1b:
77491263 8bf0            mov     esi,eax
```

Again in usermode - the "gu" command..
This gets me quickly to the return address of the current function.

```
0:000> KL4
ChildEBP RetAddr
0015f9b8 77491460 USER32!DialogBox2
0015f9e0 774914a2 USER32!InternalDialogBox+0xd0
0015fa00 774b12de USER32!DialogBoxIndirectParamAorW+0x37
0015fa24 760d1832 USER32!DialogBoxParamW+0x3f

0:000> r
eax=001a18d4 ebx=00000000 ecx=001a18d4 edx=008c0570 esi=00000001 edi=000c18fe
eip=77491244 esp=0015f9bc ebp=0015f9e0 iopl=0         nv up ei pl nz na pe nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000206
USER32!DialogBox2:
77491244 8bff            mov     edi,edi

0:000> gu
eax=00000001 ebx=00000000 ecx=0015f980 edx=77340f34 esi=00000001 edi=000c18fe
eip=77491460 esp=0015f9d0 ebp=0015f9e0 iopl=0         nv up ei pl nz na po nc
cs=001b  ss=0023  ds=0023  es=0023  fs=003b  gs=0000             efl=00000202
USER32!InternalDialogBox+0xd0:
77491460 5f              pop     edi

0:000> KL4
ChildEBP RetAddr
0015f9e0 774914a2 USER32!InternalDialogBox+0xd0
0015fa00 774b12de USER32!DialogBoxIndirectParamAorW+0x37
0015fa24 760d1832 USER32!DialogBoxParamW+0x3f
0015fa48 761ea0e5 SHELL32!SHFusionDialogBoxParam+0x32
```

DT – Dump Type… go explore this one yourself… quite handy for all kinds of things

DL – Dump a simple list – you can specify the type information via !list – another handy one.

The j command - conditional BP's sure are handy

The e* commands - great for editing code in ASM on the fly... noop out, jmp etc...

The r command - view\set registers: r EAX=00000000 , nice to see - hey what if that had succeeded (kind of dangerous used randomly)

All of the below explained in the debug help:

```
!address

!token

!sd

!pool

!pte

!heap
```

Anyway – there are a few of them…

Have fun!

spatdsg

Comments

Boo
7 May 2007 12:31 PM

Like usual I'm impressed by your wealth of knowledge.

One that I've grown attached to lately is ub. Good for when you hit an AV. You can then unassemble backwards to that point. Nice to see where that parameter came from.

ub mod!function+offset l20

SpatDSG
8 May 2007 12:31 PM

Thanks Brad… UB and UF -- two other good ones.