

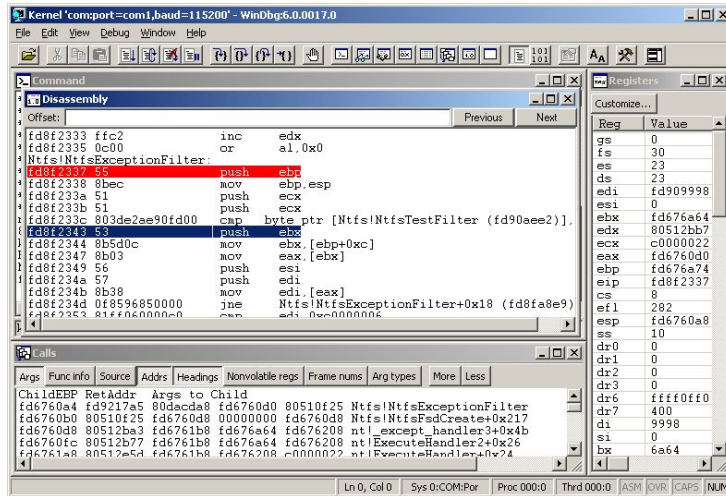


B1TFØRGE

Debug Remoto

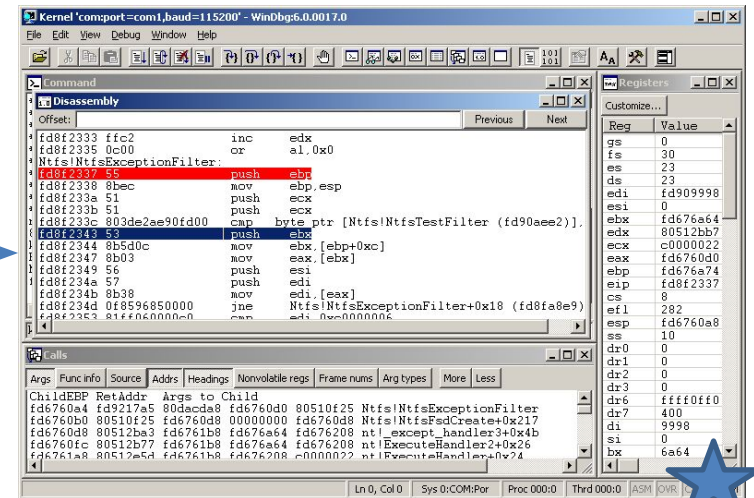
Wanderley Caloni

Windbg -remote tcp:server=vmw7, port=6666



The screenshot shows the Windbg interface with the following components:

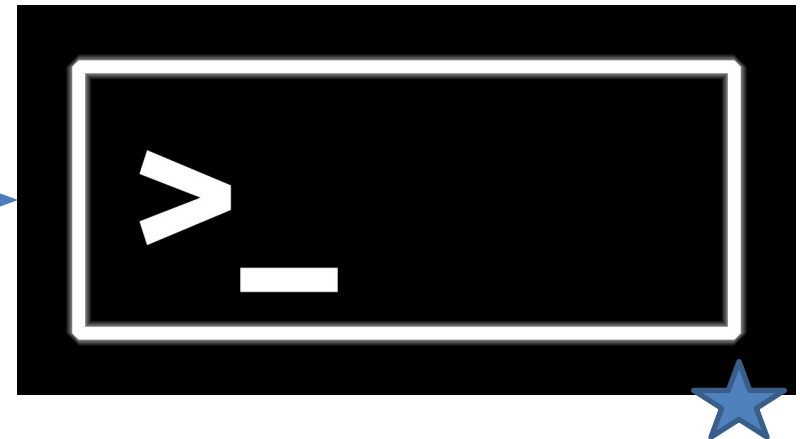
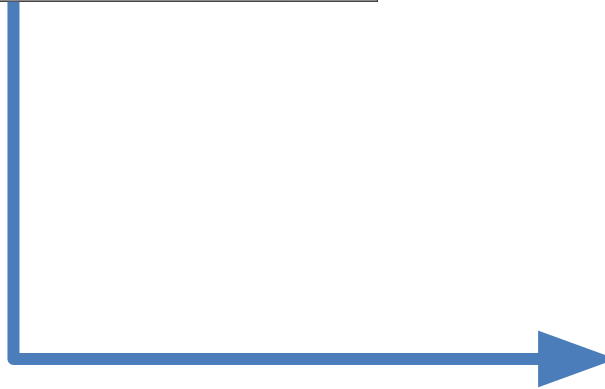
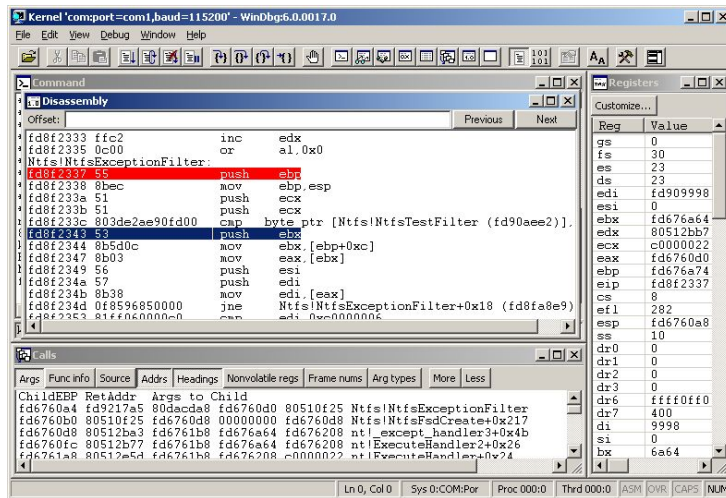
- Command Window:** Displays the command `Kernel 'comport=com1,baud=115200' - WinDbg6.0.0017.0`.
- Disassembly Window:** Shows assembly code for the `Ntfs!NtfsExceptionFilter` function. The instruction `fd8f2337 55 push ebp` is highlighted in red.
- Registers Window:** Displays the current state of CPU registers. The `eax` register contains the value `fd676a64`.
- Calls Window:** Shows the call stack, including the `ChildEBP RetAddr` and the `Ntfs!NtfsExceptionFilter` function.



This screenshot is identical to the first one, showing the same Windbg interface with the disassembly and registers windows. A blue star is placed in the bottom right corner of the interface.

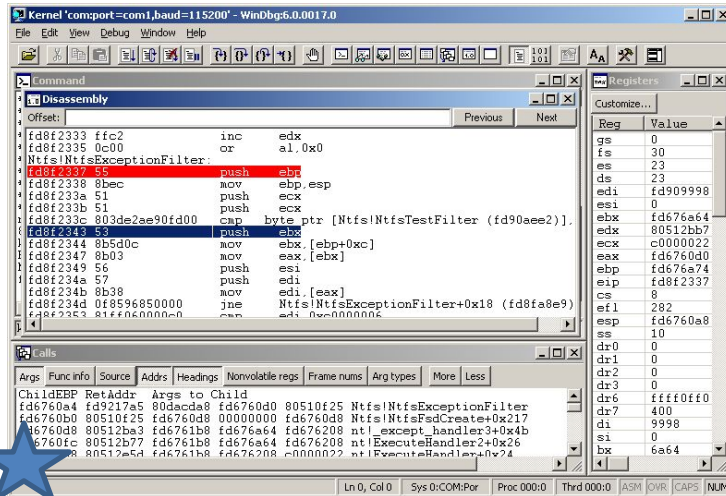
.server tcp:port=6666

Windbg -remote tcp:server=vmw7, port=6666



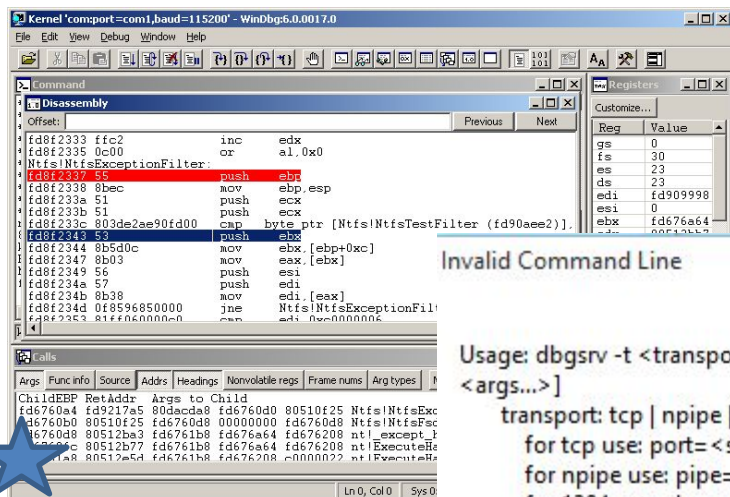
WinDbg/Nttd/Cdb -server tcp:port=6666

Windbg -remote tcp:server=vmw7, port=6666



dbgsrv -server tcp:port=6666

Windbg -remote tcp:server=vmw7, port=6666



Invalid Command Line

Usage: dbgsrv -t <transport> [-sifeo <image.ext>] [-x] [-c[s] <args...>] [-pc <args...>]

transport: tcp | npipe | ssl | spipe | 1394 | com

for tcp use: port= <socket port #>

for npipe use: pipe= <name of pipe>

for 1394 use: channel= <channel #>

for com use: port= <COM port>, baud= <baud rate>, channel= <channel #>

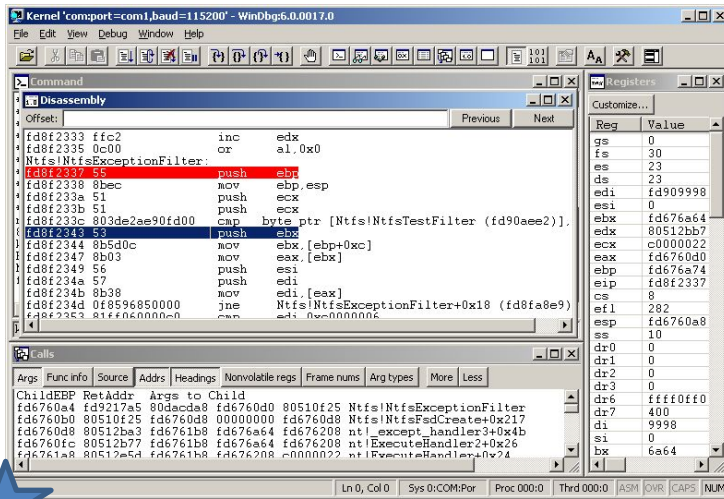
for ssl and spipe see the documentation

Example: dbgsrv -t npipe:pipe=foobar

OK

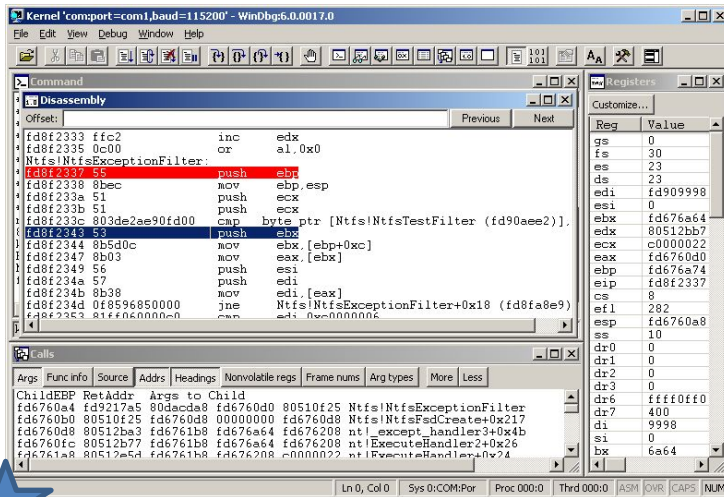
dbgsrv -t tcp:port=6666

Windbg -premove tcp:server=vmw7, port=6666

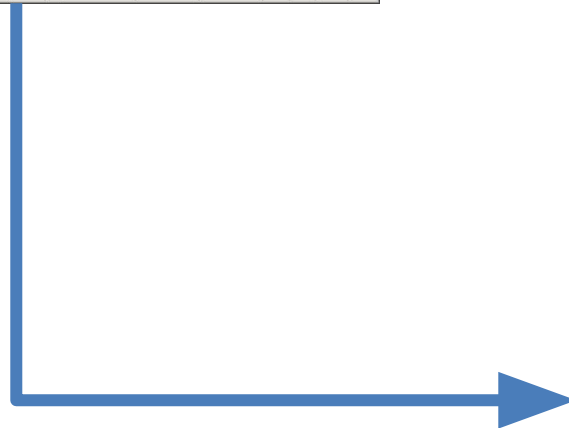


dbgsrv -t tcp:port=6666

Windbg -premove tcp:server=vmw7, port=6666

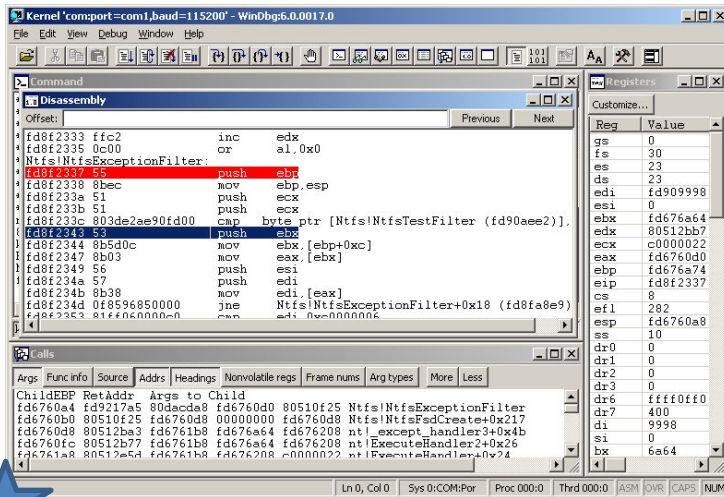


.server tcp:port=6666

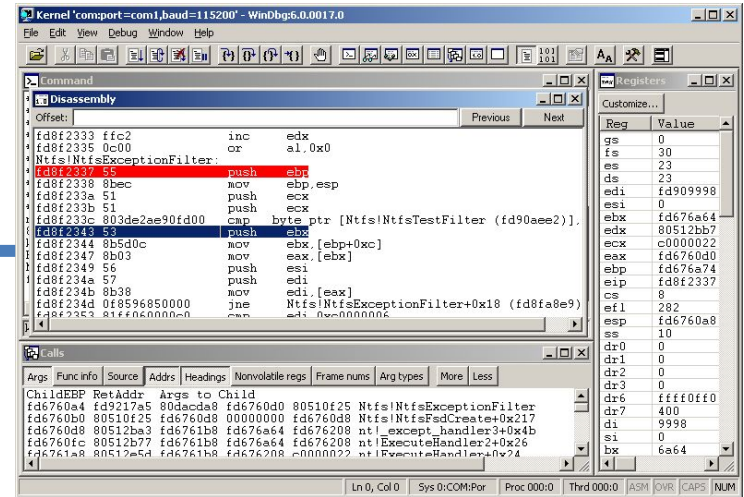


dbgsrv -t tcp:port=6666

Windbg -premove tcp:server=vmw7, port=6666



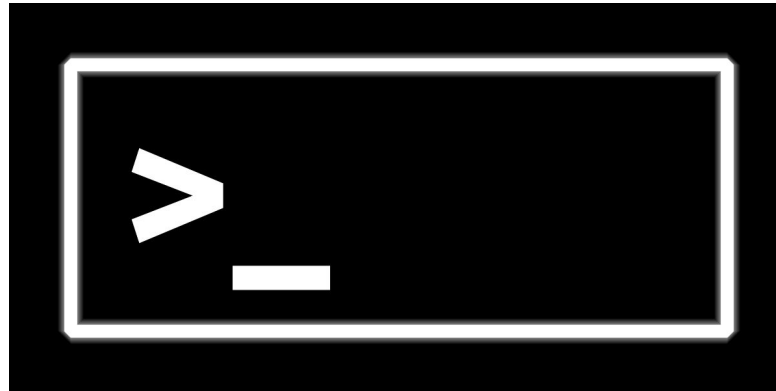
Windbg -remote tcp:server=vmw7, port=6666



.server tcp:port=6666



dbgsrv -t tcp:port=6666

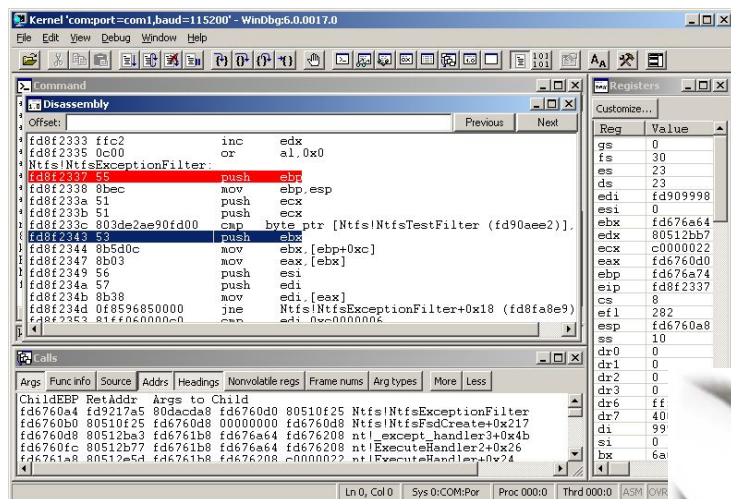


dbengprx.exe [-p] -c tcp:server=srv,port=6666 -s tcp:port=6667

dbengprx.exe [-p] -c npipe:server=srv,pipe=NPipe -s tcp:port=6667

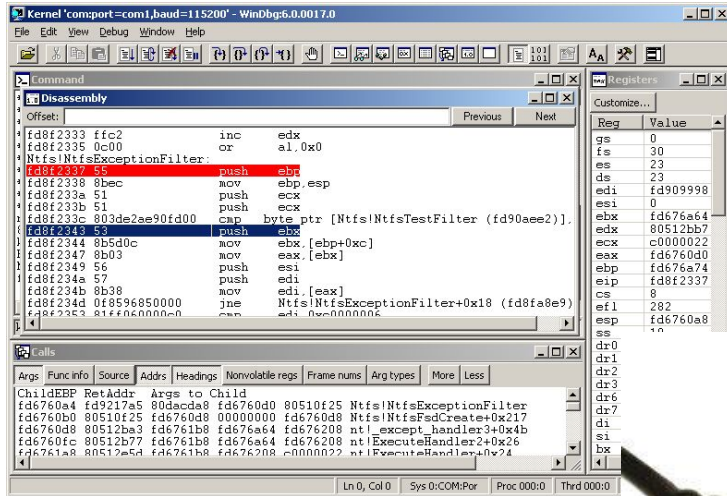
dbengprx.exe [-p] -c com:port=*COMPort*,baud=*BaudRate* -s tcp:port=6667

dbengprx.exe [-p] -c tcp:clicon=srv,port=666 -s tcp:port=6667



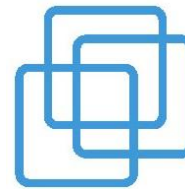
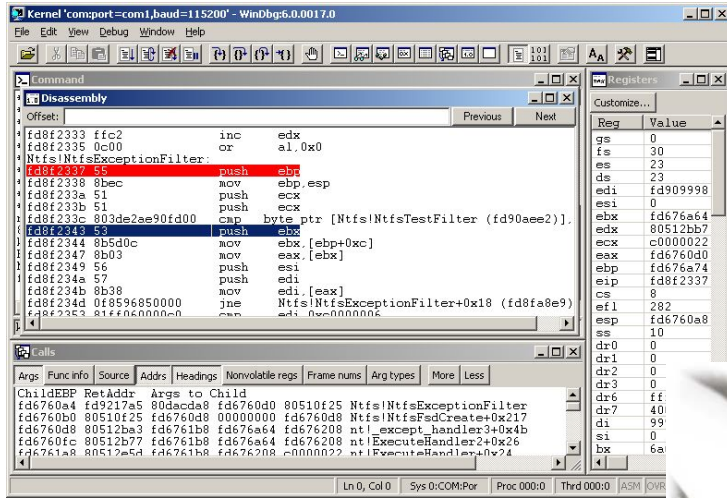
bcdedit /dbgsettings
bcdedit /copy {current}
bcdedit set debug on

windbg.exe -k com:port=1,baud=115200,reset=0,reconnect -b



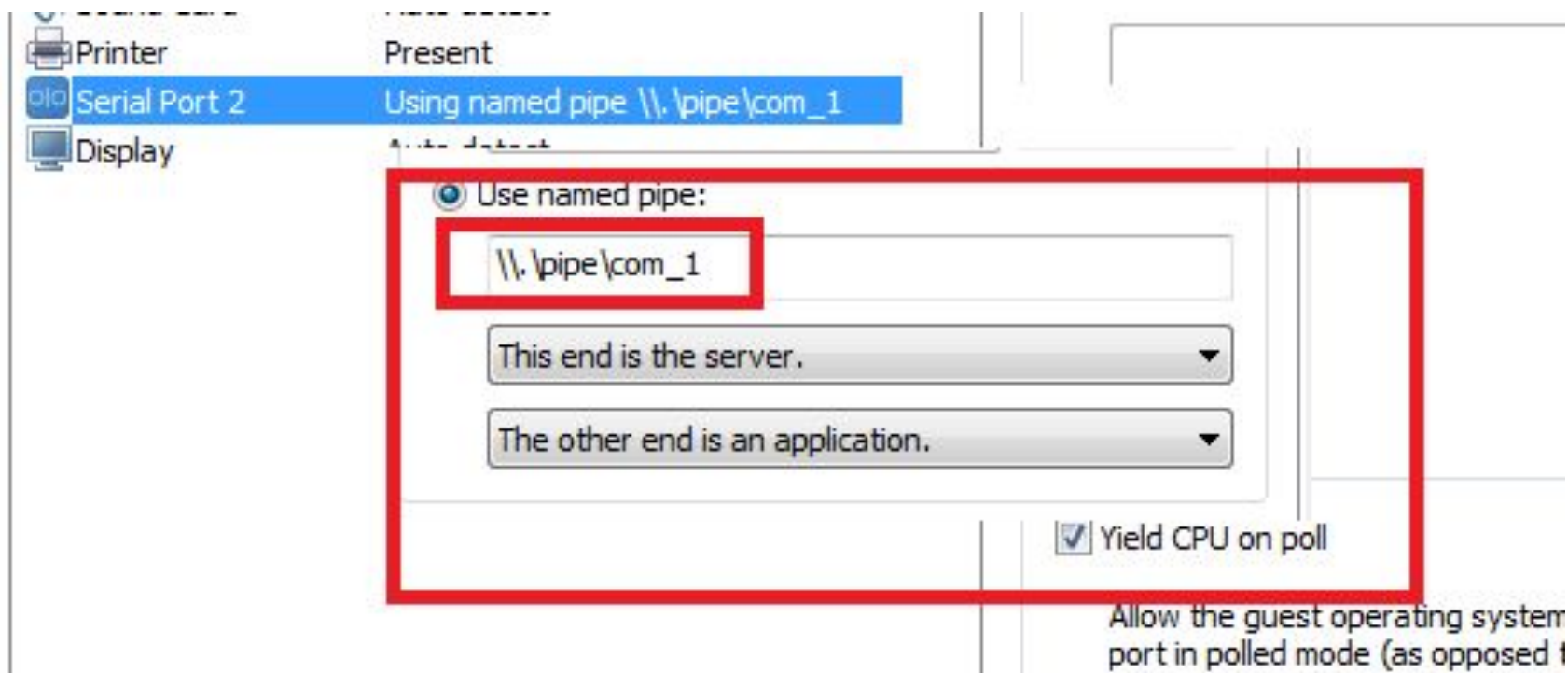
bcdedit /dbgsettings
bcdedit /copy {current}
bcdedit set debug on

windbg.exe -k usb2:targetname=USBString -b



vmware™

windbg.exe -k com:pipe,port=\\.\pipe\com_1, resets=0, reconnect -b



```
windbg.exe -k com:pipe,port=\\.\\pipe\\com_1, resets=0, reconnect -b
```

A screenshot of a Windows debugger configuration dialog box. The dialog has a light gray background and a standard Windows window border. It contains several checkboxes and text input fields. The checkboxes are arranged in two columns. The first column has three checkboxes: 'Enable heap tagging by DLL', 'Load image using large pages if possible', and 'Debugger:'. The second column has two checkboxes: 'Disable protected DLL verification' and 'Ignore asserts'. The 'Debugger:' checkbox is checked, and its corresponding text input field contains the path 'c:\tools\dbgtools\windbg.exe'. Below the 'Debugger:' checkbox is another checkbox labeled 'Stack Backtrace: (Megs)' with an empty text input field next to it. At the bottom right of the dialog are three buttons: 'OK', 'Cancel', and 'Apply'.

☐ Enable heap tagging by DLL

☐ Disable protected DLL verification

☐ Load image using large pages if possible

☒ Debugger: c:\tools\dbgtools\windbg.exe

☐ Stack Backtrace: (Megs)

OK Cancel Apply

GLOBAL FLAGS

PROCESS

DEBUGGER

USER

KERNEL

CABLE

DEBUGGER

