SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)

/dev/random - Useful WinDbg Resources

MARCH 31ST, 2019

2 Minute Read

What Is This?

This is an ever growing collection of resources that I found to be useful while researching and learning about WinDbg. I have created this blog post as a place to keep track of resources.

Workspace Settings

Customizing your WinDbg Workspace and Color Scheme

zach burlingam windbg workspace key backup

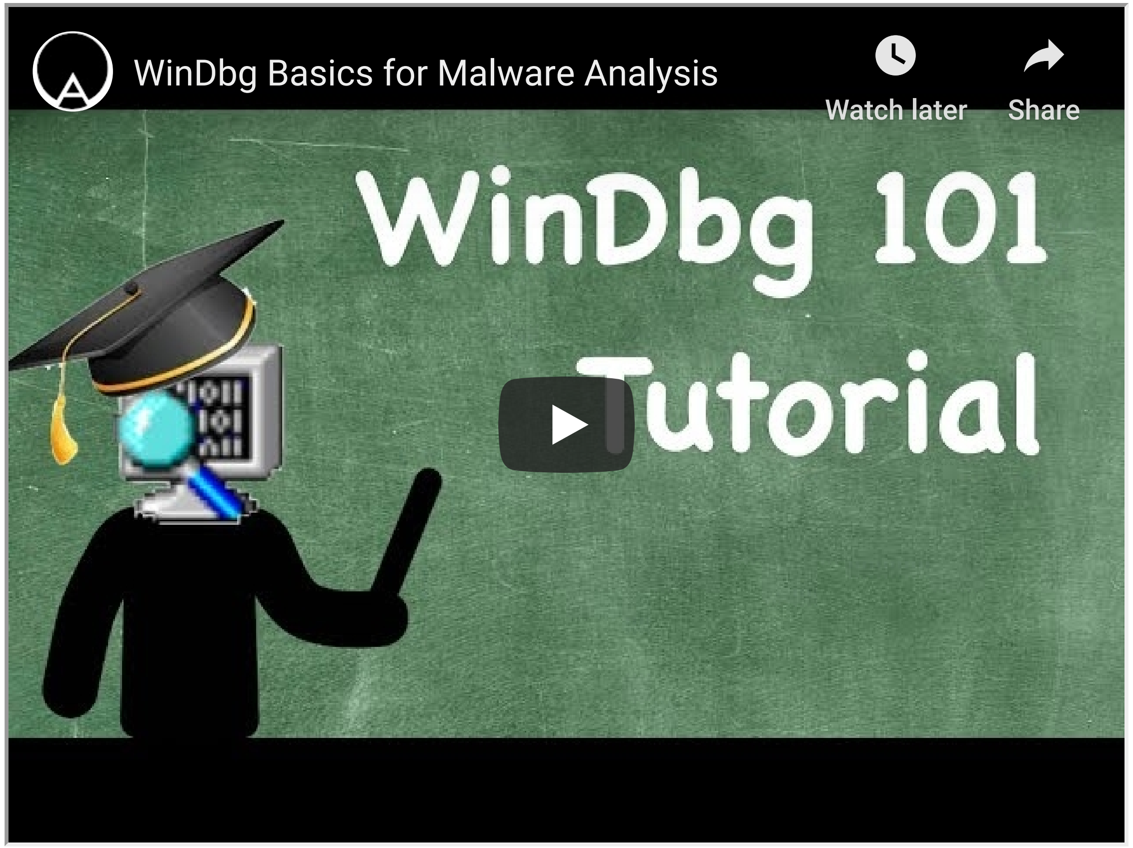
YouTube Videos

OALabs

WinDbg Basics for Malware Analysis

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)



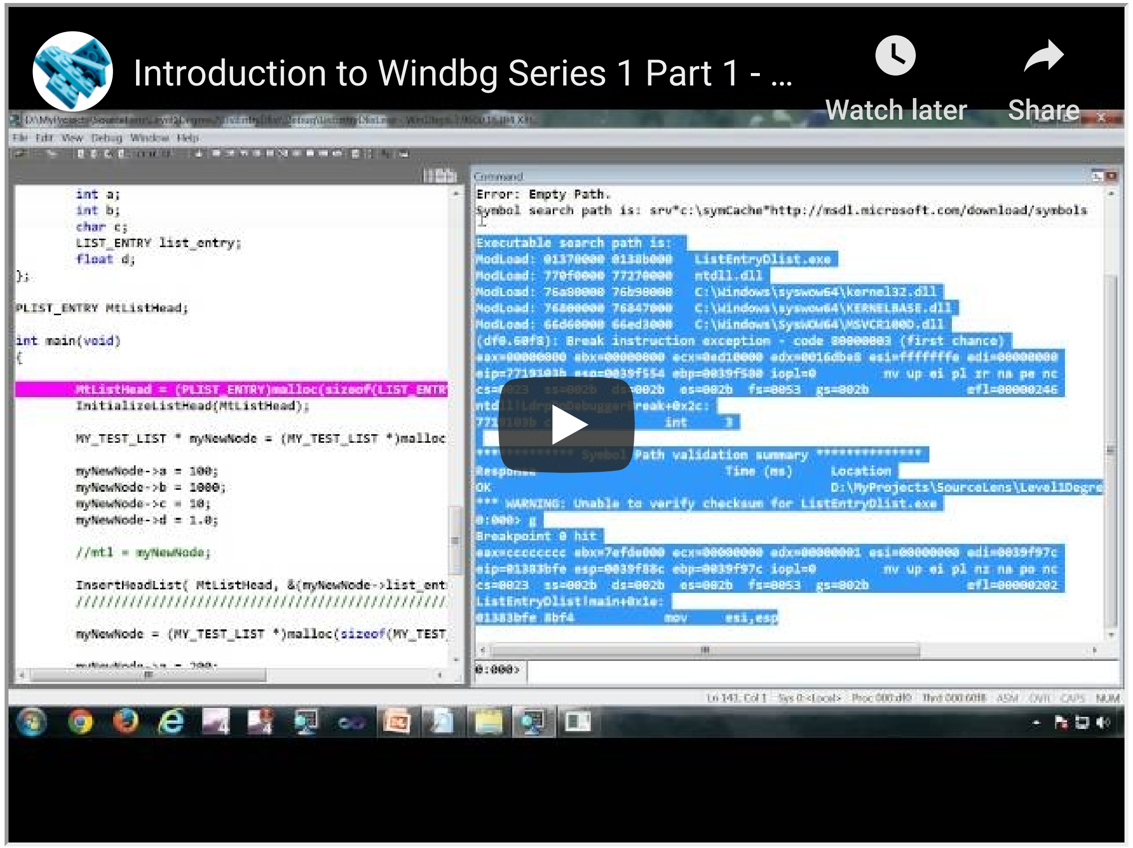
TheSourceLens

Part 01: THE Debugger

SECURITY CHOPS

growing my chops in cybersecurity

(all opinions are my own and not the views of my employer)



Introduction to Windbg Series 1 Part 1 - ...
Watch later
Share
File Edit View Debug Window Help
Command

```
        int a;
        int b;
        char c;
        LIST_ENTRY list_entry;
        float d;
};

PLIST_ENTRY MtListHead;

int main(void)
{

        MtListHead = (PLIST_ENTRY)malloc(sizeof(LIST_ENTR
        InitializeListHead(MtListHead);

        MY_TEST_LIST * myNewNode = (MY_TEST_LIST *)malloc

        myNewNode->a = 100;
        myNewNode->b = 1000;
        myNewNode->c = 10;
        myNewNode->d = 1.0;

        //mtl = myNewNode;

        InsertHeadList( MtListHead, &(myNewNode->list_ent
        ////////////////////////////////////////////////

        myNewNode = (MY_TEST_LIST *)malloc(sizeof(MY_TEST_
```

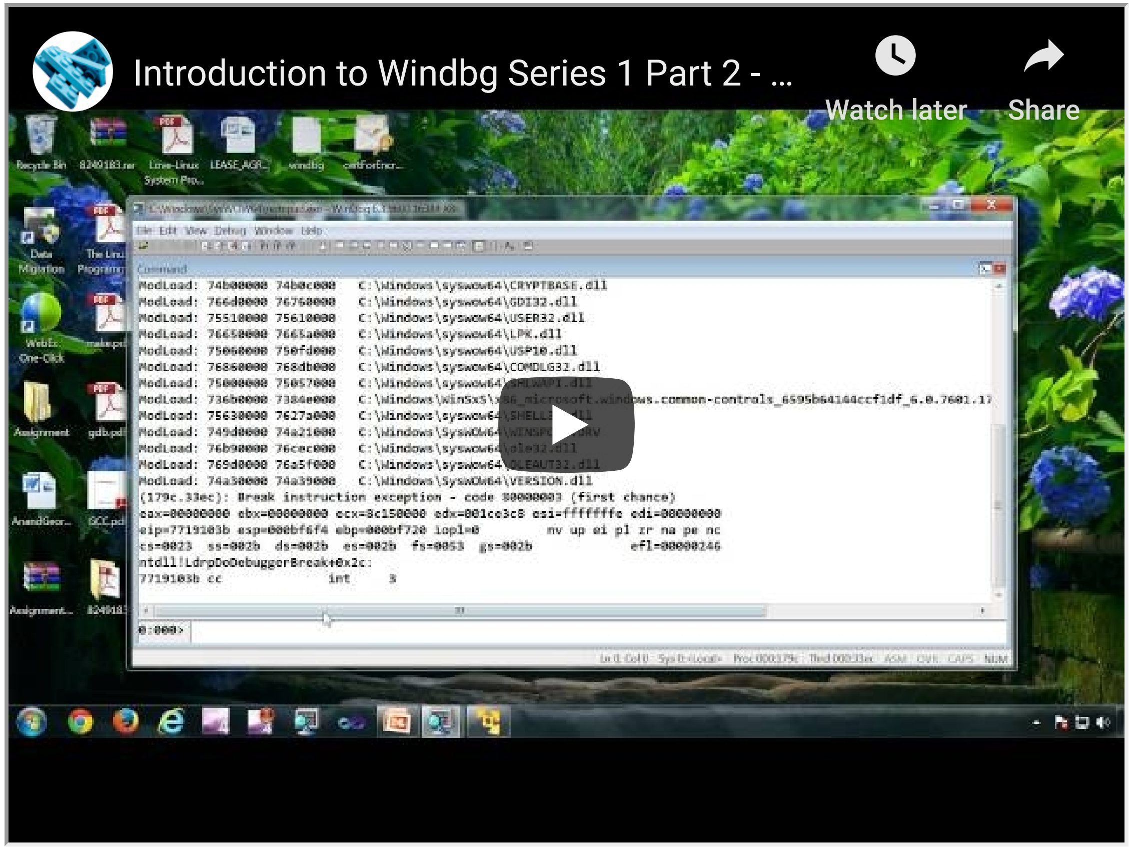
```
Error: Empty Path.
Symbol search path is: srv*c:\symCache*http://msdl.microsoft.com/download/symbols

Executable search path is:
ModLoad: 01370000 0138b000   ListEntryDlist.exe
ModLoad: 770f0000 77270000   ntdll.dll
ModLoad: 76a80000 76b90000   C:\Windows\syswow64\kernel32.dll
ModLoad: 76800000 76847000   C:\Windows\syswow64\KERNELBASE.dll
ModLoad: 66d60000 66ed3000   C:\Windows\SysWOW64\MSVCR100D.dll
(df0.60f8): Break instruction exception - code 80000003 (first chance)
eax=00000000 ebx=00000000 ecx=0ad10000 edx=0016dbe8 esi=fffffffe edi=00000000
eip=7719103b esp=0039f554 ebp=0039f580 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
ntdll!LdrpDoDebuggerBreak+0x2c:
7719103b cc              int     3

************** Symbol Path validation summary **************
Response                         Time (ms)     Location
OK                                             D:\MyProjects\SourceLens\Level1Degre
*** WARNING: Unable to verify checksum for ListEntryDlist.exe
0:000> g
Breakpoint 0 hit
eax=cccccccc ebx=7efde000 ecx=00000000 edx=00000001 esi=00000000 edi=0039f97c
eip=01383bfe esp=0039f88c ebp=0039f97c iopl=0         nv up ei pl nz na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000202
ListEntryDlist!main+0x1e:
01383bfe 8bf4            mov     esi,esp

0:000>
```

Ln 141, Col 1 Sys 0:<Local> Proc 000:df0 Thrd 000:60f8 ASM OVR CAPS NUM

Part 02: Different Modes Of Operations of Windbg

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)



Introduction to Windbg Series 1 Part 2 - ...
Watch later
Share
File Edit View Debug Window Help
Command

```
ModLoad: 74b00000 74b0c000   C:\Windows\syswow64\CRYPTBASE.dll
ModLoad: 766d0000 76760000   C:\Windows\syswow64\GDI32.dll
ModLoad: 75510000 75610000   C:\Windows\syswow64\USER32.dll
ModLoad: 76650000 7665a000   C:\Windows\syswow64\LPK.dll
ModLoad: 75060000 750fd000   C:\Windows\syswow64\USP10.dll
ModLoad: 76860000 768db000   C:\Windows\syswow64\COMDLG32.dll
ModLoad: 75000000 75057000   C:\Windows\syswow64\SHLWAPI.dll
ModLoad: 736b0000 7384e000   C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.7601.17
ModLoad: 75630000 7627a000   C:\Windows\syswow64\SHELL32.dll
ModLoad: 749d0000 74a21000   C:\Windows\SysWOW64\WINSPOOL.DRV
ModLoad: 76b90000 76cec000   C:\Windows\syswow64\ole32.dll
ModLoad: 769d0000 76a5f000   C:\Windows\syswow64\OLEAUT32.dll
ModLoad: 74a30000 74a39000   C:\Windows\SysWOW64\VERSION.dll
(179c.33ec): Break instruction exception - code 80000003 (first chance)
eax=00000000 ebx=00000000 ecx=8c150000 edx=001ce3c8 esi=fffffffe edi=00000000
eip=7719103b esp=000bf6f4 ebp=000bf720 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
ntdll!LdrpDoDebuggerBreak+0x2c:
7719103b cc              int     3
0:000>
```

Part 03: Introduction To debug Symbols

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)

Introduction to Windbg Series 1 Part 3 - I...

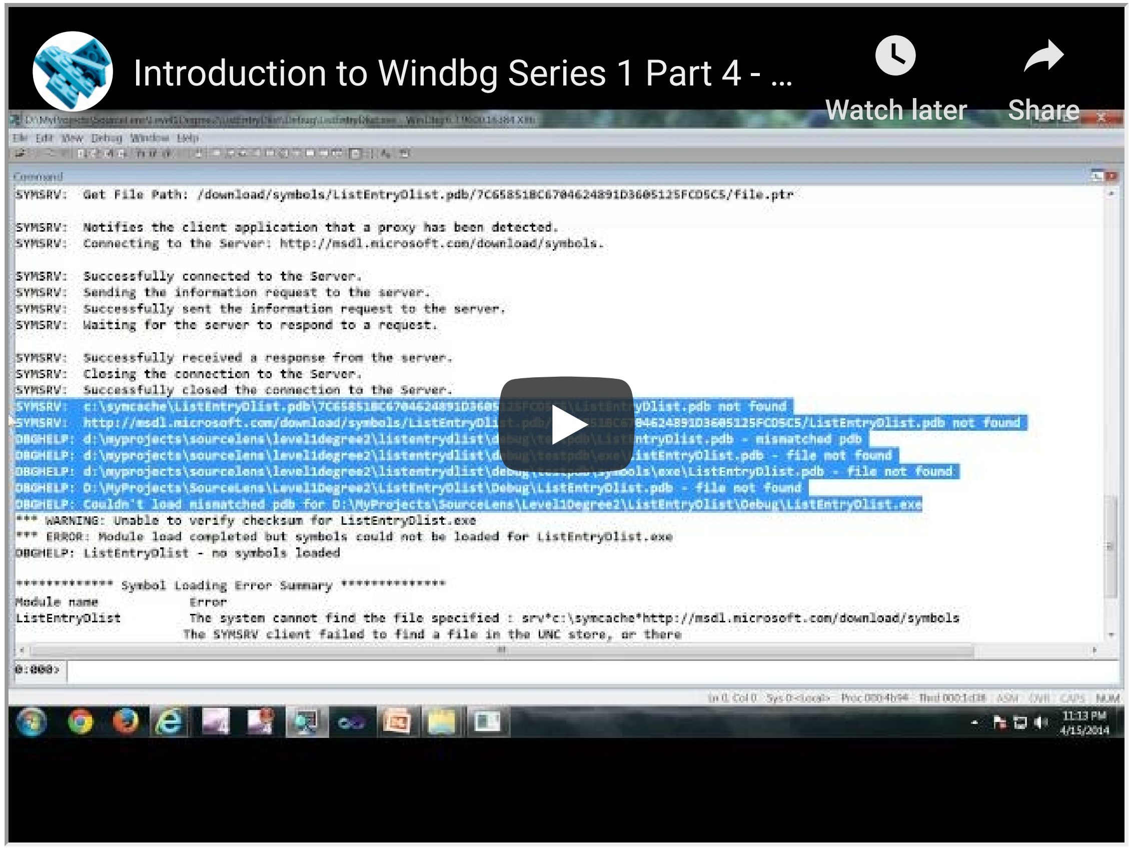
```
   0 System Process
   4 System
 320 smss.exe
 452 csrss.exe
 520 wininit.exe
 540 csrss.exe
 576 services.exe
 596 lsass.exe
 604 lsm.exe
 696 winlogon.exe
 760 svchost.exe
 828 nvvsvc.exe
 852 nvSCPAPISvr.exe
 896 svchost.exe
 996 svchost.exe
 152 svchost.exe
```

Part 04: Troubleshooting Symbols mismatch

SECURITY CHOPS

growing my chops in cybersecurity

(all opinions are my own and not the views of my employer)

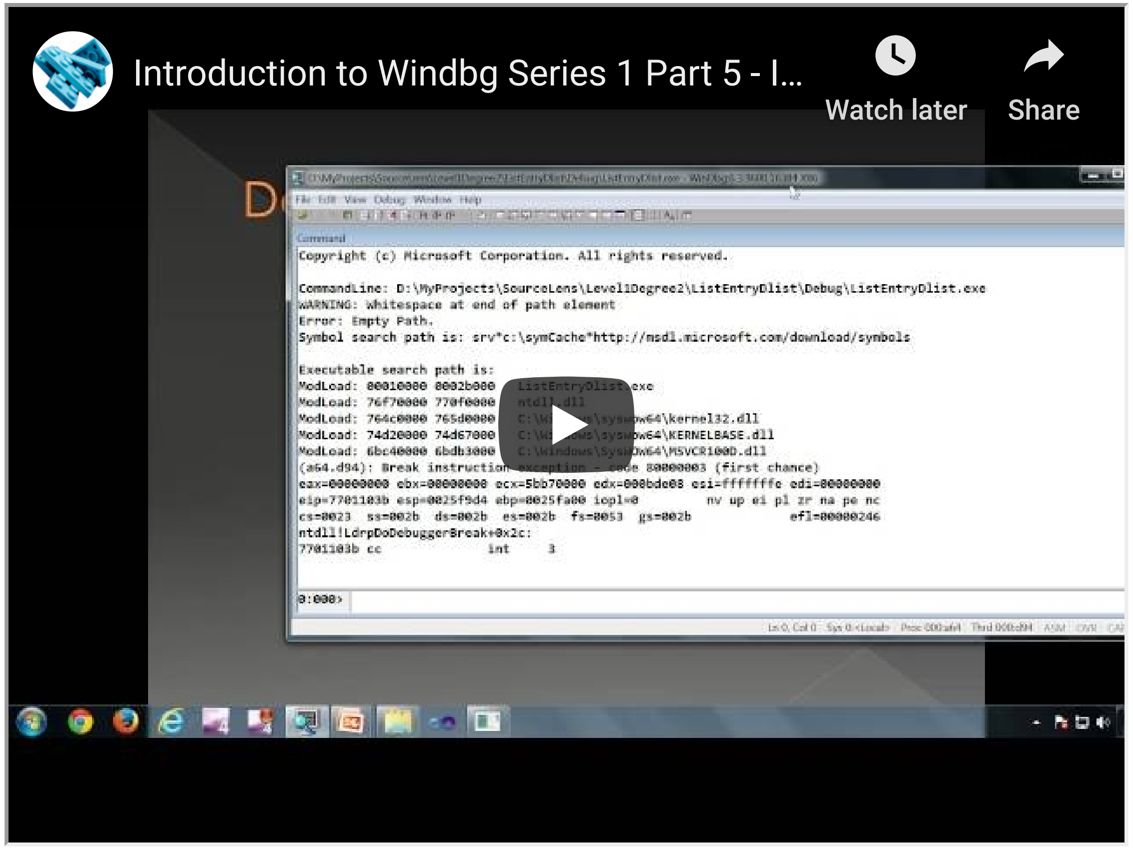

Introduction to Windbg Series 1 Part 4 - ...
Watch later
Share
File Edit View Debug Window Help
Command

```
SYMSRV:  Get File Path: /download/symbols/ListEntryDlist.pdb/7C65851BC6704624891D3605125FCD5C5/file.ptr

SYMSRV:  Notifies the client application that a proxy has been detected.
SYMSRV:  Connecting to the Server: http://msdl.microsoft.com/download/symbols.

SYMSRV:  Successfully connected to the Server.
SYMSRV:  Sending the information request to the server.
SYMSRV:  Successfully sent the information request to the server.
SYMSRV:  Waiting for the server to respond to a request.

SYMSRV:  Successfully received a response from the server.
SYMSRV:  Closing the connection to the Server.
SYMSRV:  Successfully closed the connection to the Server.
SYMSRV:  c:\symcache\ListEntryDlist.pdb\7C65851BC6704624891D3605125FCD5C5\ListEntryDlist.pdb not found
SYMSRV:  http://msdl.microsoft.com/download/symbols/ListEntryDlist.pdb/7C65851BC6704624891D3605125FCD5C5/ListEntryDlist.pdb not found
DBGHELP: d:\myprojects\sourcelens\level1degree2\listentrydlist\debug\testpdb\ListEntryDlist.pdb - mismatched pdb
DBGHELP: d:\myprojects\sourcelens\level1degree2\listentrydlist\debug\testpdb\exe\ListEntryDlist.pdb - file not found
DBGHELP: d:\myprojects\sourcelens\level1degree2\listentrydlist\debug\testpdb\symbols\exe\ListEntryDlist.pdb - file not found
DBGHELP: D:\MyProjects\SourceLens\Level1Degree2\ListEntryDlist\Debug\ListEntryDlist.pdb - file not found
DBGHELP: Couldn't load mismatched pdb for D:\MyProjects\SourceLens\Level1Degree2\ListEntryDlist\Debug\ListEntryDlist.exe
*** WARNING: Unable to verify checksum for ListEntryDlist.exe
*** ERROR: Module load completed but symbols could not be loaded for ListEntryDlist.exe
DBGHELP: ListEntryDlist - no symbols loaded

************* Symbol Loading Error Summary **************
Module name            Error
ListEntryDlist         The system cannot find the file specified : srv*c:\symcache*http://msdl.microsoft.com/download/symbols
                      The SYMSRV client failed to find a file in the UNC store, or there

0:000>
```

Part 05: Introduction to debugger Commands

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)


Introduction to Windbg Series 1 Part 5 - I...
Watch later
Share
File Edit View Debug Window Help
Command

```
Copyright (c) Microsoft Corporation. All rights reserved.

CommandLine: D:\MyProjects\SourceLens\Level1Degree2\ListEntryDlist\Debug\ListEntryDlist.exe
WARNING: Whitespace at end of path element
Error: Empty Path.
Symbol search path is: srv*c:\symCache*http://msdl.microsoft.com/download/symbols

Executable search path is:
ModLoad: 00010000 0002b000   ListEntryDlist.exe
ModLoad: 76f70000 770f0000   ntdll.dll
ModLoad: 764c0000 765d0000   C:\Windows\syswow64\kernel32.dll
ModLoad: 74d20000 74d67000   C:\Windows\syswow64\KERNELBASE.dll
ModLoad: 6bc40000 6bdb3000   C:\Windows\SysWOW64\MSVCR100D.dll
(a64.d94): Break instruction exception - code 80000003 (first chance)
eax=00000000 ebx=00000000 ecx=5bb70000 edx=000bde08 esi=fffffffe edi=00000000
eip=7701103b esp=0025f9d4 ebp=0025fa00 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
ntdll!LdrpDoDebuggerBreak+0x2c:
7701103b cc              int     3

0:000>
```

Part 06: Kernel Debugging With VmPlayer

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)

Introduction to Windbg Series 1 Part 6 - ...
Watch later
Share
Setup
Host OS
Vm Player
Windbg in kernel debuggin mode.
arget Windows OS

Part 07: Physical Machine Kernel Debugging With Network Cable

SECURITY CHOPS

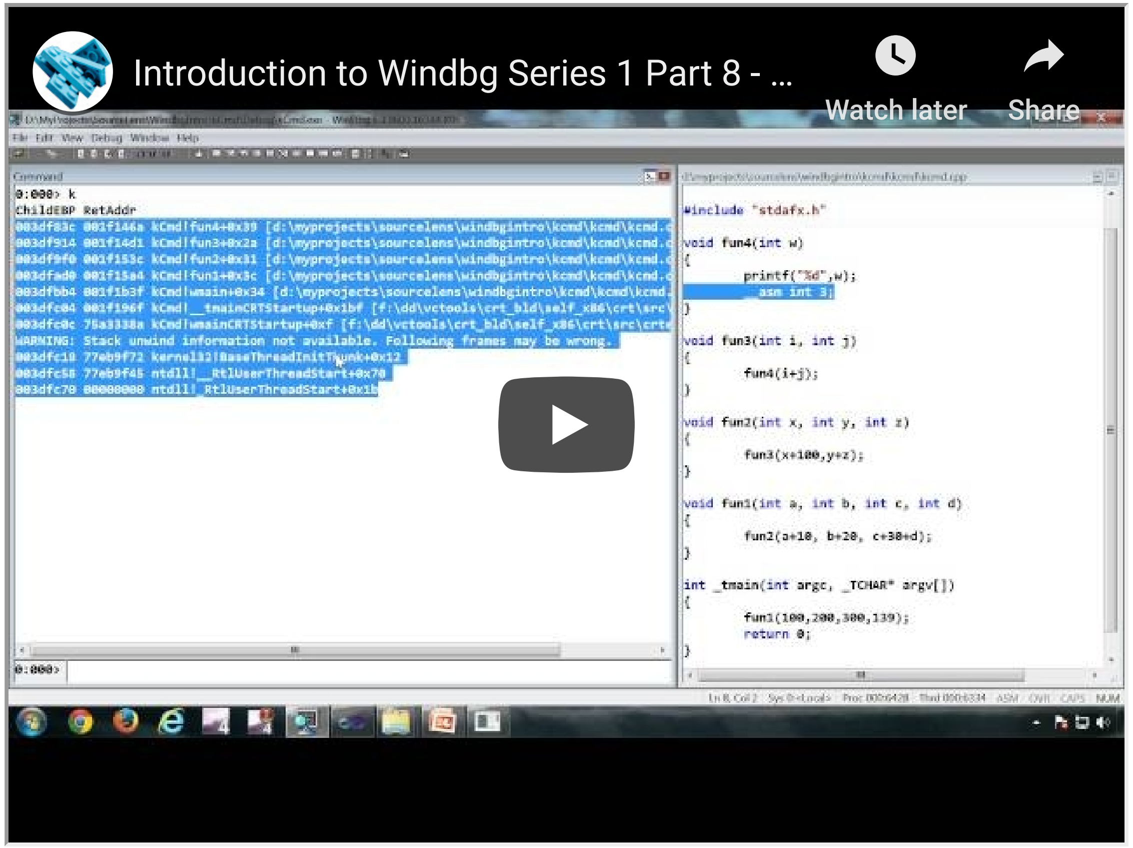
growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)

Introduction to Windbg Series 1 Part 7 - ...

Part 08: Commands k for callstack or stackback trace

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)


Introduction to Windbg Series 1 Part 8 - ...
Watch later
Share
File Edit View Debug Window Help
Command

```
0:000> k
ChildEBP RetAddr
003df83c 001f146a kCmd!fun4+0x39 [d:\myprojects\sourcelens\windbgintro\kcmd\kcmd\kcmd.c
003df914 001f14d1 kCmd!fun3+0x2a [d:\myprojects\sourcelens\windbgintro\kcmd\kcmd\kcmd.c
003df9f0 001f153c kCmd!fun2+0x31 [d:\myprojects\sourcelens\windbgintro\kcmd\kcmd\kcmd.c
003dfad0 001f15a4 kCmd!fun1+0x3c [d:\myprojects\sourcelens\windbgintro\kcmd\kcmd\kcmd.c
003dfbb4 001f1b3f kCmd!wmain+0x34 [d:\myprojects\sourcelens\windbgintro\kcmd\kcmd\kcmd.
003dfc04 001f196f kCmd!__tmainCRTStartup+0x1bf [f:\dd\vctools\crt_bld\self_x86\crt\src\
003dfc0c 75a3338a kCmd!wmainCRTStartup+0xf [f:\dd\vctools\crt_bld\self_x86\crt\src\crte
WARNING: Stack unwind information not available. Following frames may be wrong.
003dfc18 77eb9f72 kernel32!BaseThreadInitThunk+0x12
003dfc58 77eb9f45 ntdll!__RtlUserThreadStart+0x70
003dfc70 00000000 ntdll!_RtlUserThreadStart+0x1b
```

```
0:000>
```

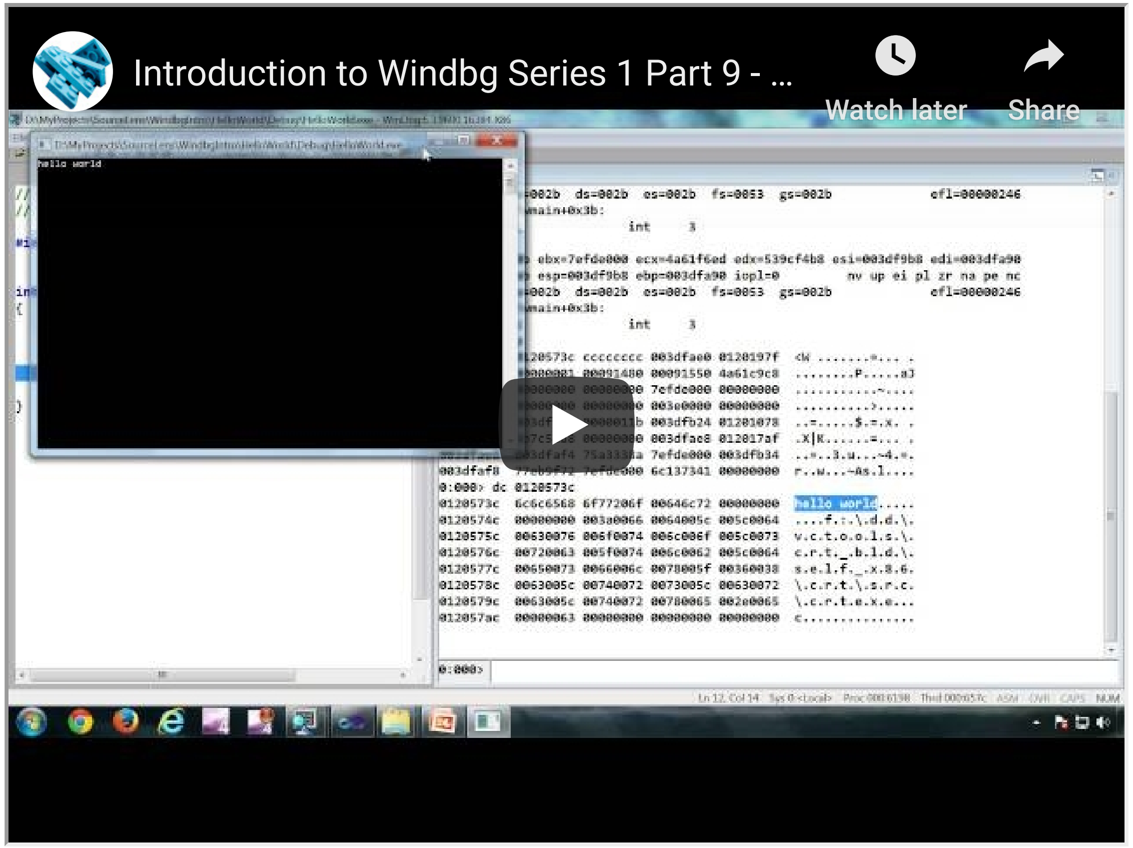
```
#include "stdafx.h"

void fun4(int w)
{
        printf("%d",w);
          asm int 3;
}

void fun3(int i, int j)
{
        fun4(i+j);
}

void fun2(int x, int y, int z)
{
        fun3(x+100,y+z);
}

void fun1(int a, int b, int c, int d)
{
        fun2(a+10, b+20, c+30+d);
}

int _tmain(int argc, _TCHAR* argv[])
{
        fun1(100,200,300,139);
        return 0;
}
```

Part 09: Commands r for register d for dump memory

SECURITY CHOPS

growing my chops in cybersecurity

(all opinions are my own and not the views of my employer)



```
hello world
```

```
=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
main+0x3b:
              int     3

 ebx=7efde000 ecx=4a61f6ed edx=539cf4b8 esi=003df9b8 edi=003dfa90
 esp=003df9b8 ebp=003dfa90 iopl=0         nv up ei pl zr na pe nc
=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
main+0x3b:
              int     3

120573c cccccccc 003dfae0 0120197f  <W .......=... .
0000001 00091480 00091550 4a61c9c8  ........P.....aJ
0000000 00000000 7efde000 00000000  ...........~....
0000000 00000000 003e0000 00000000  ..........>.....
03df    0000011b 003dfb24 01201078  ..=.....$.=.x. .
 7c5    00000000 003dfac8 012017af  .X|K......=... .
        003dfaf4 75a3338a 7efde000 003dfb34  ..=..3.u...~4.=.
003dfaf8                    6c137341 00000000  r..w...~As.l....
0:000> dc 0120573c
0120573c  6c6c6568 6f77206f 00646c72 00000000  hello world.....
0120574c  00000000 003a0066 0064005c 005c0064  ....f.:.\.d.d.\.
0120575c  00630076 006f0074 006c006f 005c0073  v.c.t.o.o.l.s.\.
0120576c  00720063 005f0074 006c0062 005c0064  c.r.t._.b.l.d.\.
0120577c  00650073 0066006c 0078005f 00360038  s.e.l.f._.x.8.6.
0120578c  0063005c 00740072 0073005c 00630072  \.c.r.t.\.s.r.c.
0120579c  0063005c 00740072 00780065 002e0065  \.c.r.t.e.x.e...
012057ac  00000063 00000000 00000000 00000000  c...............
```

```
0:000>
```

Part 10: Commands dv and .frame

SECURITY
CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)

Introduction to Windbg Series 1 Part 10 ...

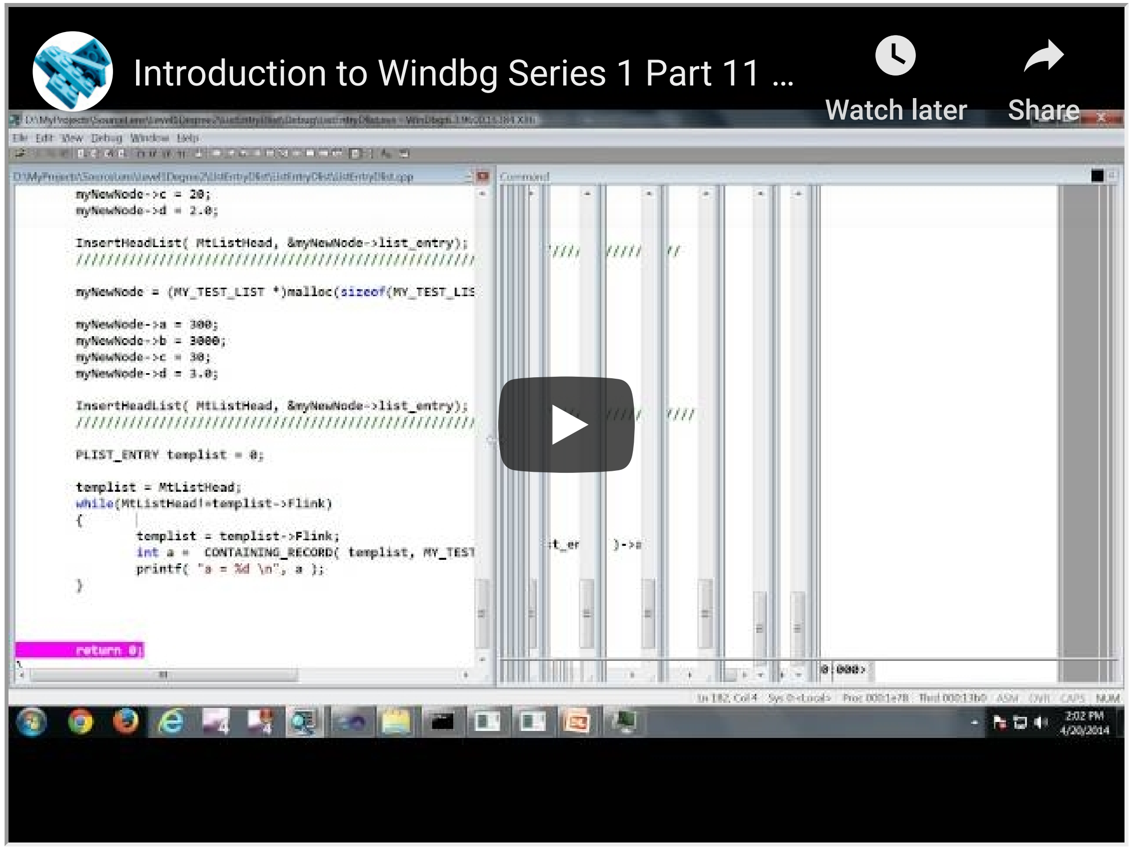
dv

- Dump locals and aruments passed for the current frame.

Part 11: Command dt - dump type

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)



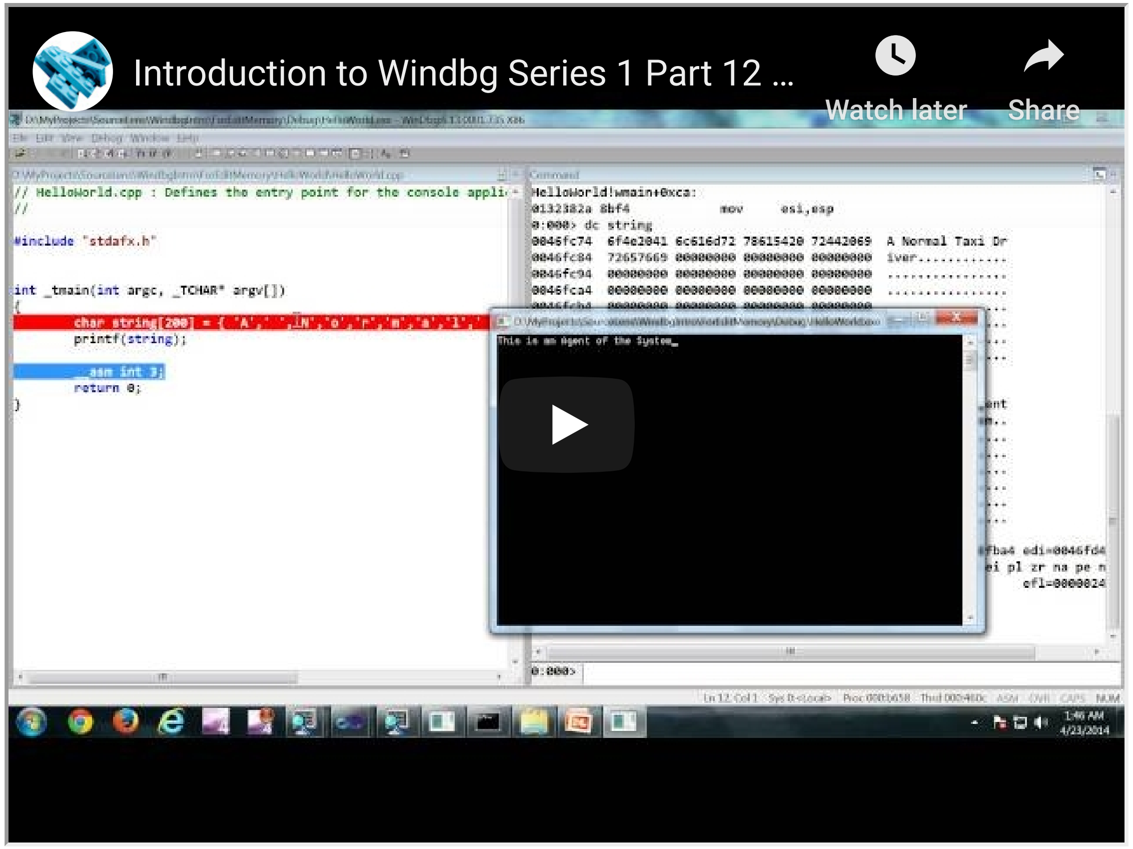
Introduction to Windbg Series 1 Part 11 ...
Watch later
Share
File Edit View Debug Window Help
Command

```
        myNewNode->c = 20;
        myNewNode->d = 2.0;

        InsertHeadList( MtListHead, &myNewNode->list_entry);
        ////////////////////////////////////////////////////////////

        myNewNode = (MY_TEST_LIST *)malloc(sizeof(MY_TEST_LIS

        myNewNode->a = 300;
        myNewNode->b = 3000;
        myNewNode->c = 30;
        myNewNode->d = 3.0;

        InsertHeadList( MtListHead, &myNewNode->list_entry);
        ////////////////////////////////////////////////////////////

        PLIST_ENTRY templist = 0;

        templist = MtListHead;
        while(MtListHead!=templist->Flink)
        {
                templist = templist->Flink;
                int a =  CONTAINING_RECORD( templist, MY_TEST
                printf( "a = %d \n", a );
        }

        return 0;
```

Part 12: Command e - edit memory

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)



```
// HelloWorld.cpp : Defines the entry point for the console appli
//

#include "stdafx.h"


int _tmain(int argc, _TCHAR* argv[])
{
	char string[200] = { 'A',' ','N','o','r','m','a','l',
	printf(string);

	__asm int 3;
	return 0;
}
```

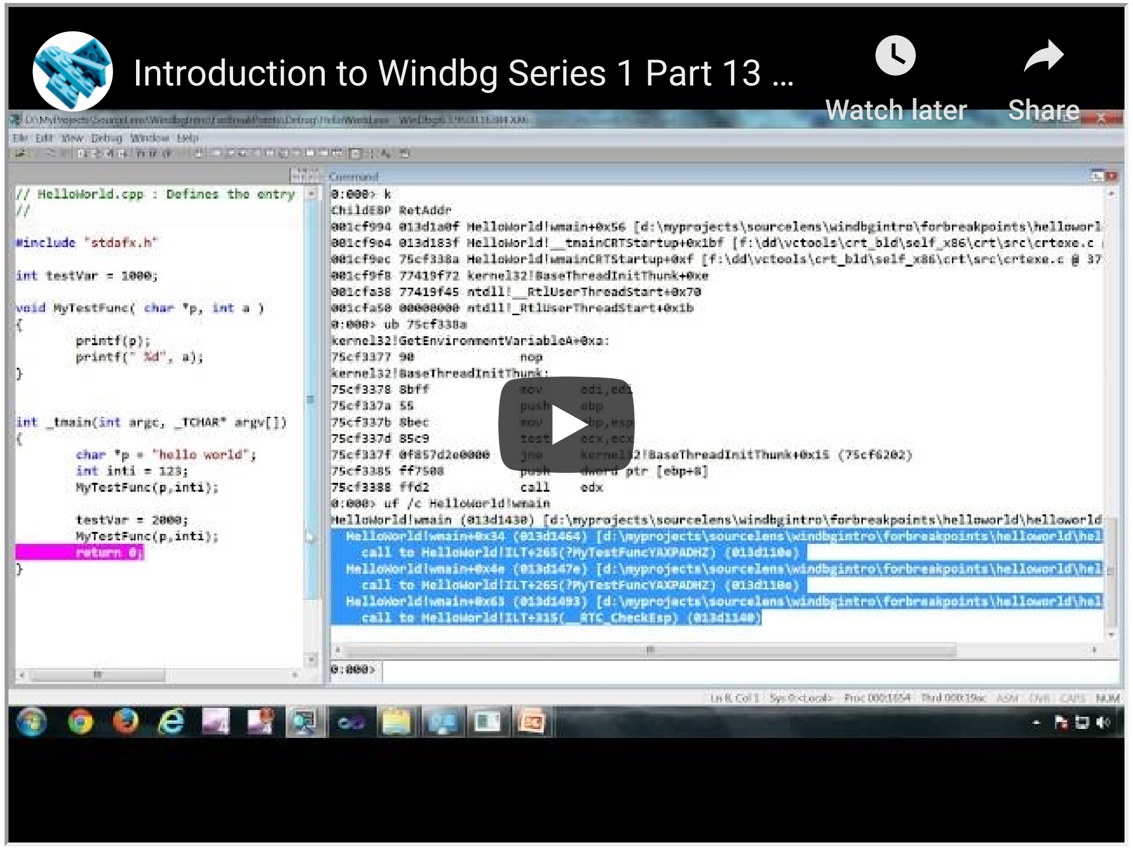
```
HelloWorld!wmain+0xca:
0132382a 8bf4            mov     esi,esp
0:000> dc string
0046fc74  6f4e2041 6c616d72 78615420 72442069  A Normal Taxi Dr
0046fc84  72657669 00000000 00000000 00000000  iver............
0046fc94  00000000 00000000 00000000 00000000  ................
0046fca4  00000000 00000000 00000000 00000000  ................
```

```
This is an Agent of the System_
```

Part 13: Unassemble code

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)


Introduction to Windbg Series 1 Part 13 ...
Watch later
Share
File Edit View Debug Window Help
Command

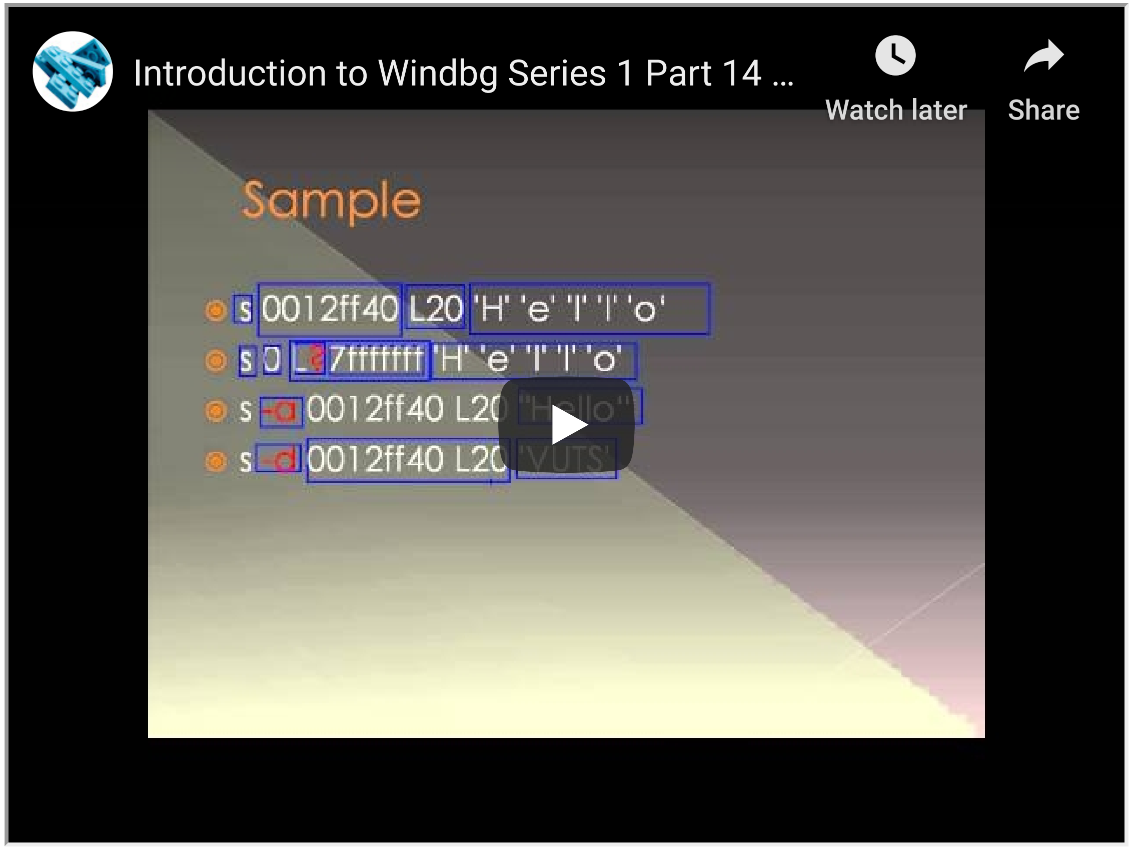
```
// HelloWorld.cpp : Defines the entry
//

#include "stdafx.h"

int testVar = 1000;

void MyTestFunc( char *p, int a )
{
	printf(p);
	printf(" %d", a);
}


int _tmain(int argc, _TCHAR* argv[])
{
	char *p = "hello world";
	int inti = 123;
	MyTestFunc(p,inti);

	testVar = 2000;
	MyTestFunc(p,inti);
	return 0;
}
```

```
0:000> k
ChildEBP RetAddr
001cf994 013d1a0f HelloWorld!wmain+0x56 [d:\myprojects\sourcelens\windbgintro\forbreakpoints\helloworl
001cf9e4 013d183f HelloWorld!__tmainCRTStartup+0x1bf [f:\dd\vctools\crt_bld\self_x86\crt\src\crtexe.c
001cf9ec 75cf338a HelloWorld!wmainCRTStartup+0xf [f:\dd\vctools\crt_bld\self_x86\crt\src\crtexe.c @ 37
001cf9f8 77419f72 kernel32!BaseThreadInitThunk+0xe
001cfa38 77419f45 ntdll!__RtlUserThreadStart+0x70
001cfa50 00000000 ntdll!_RtlUserThreadStart+0x1b
0:000> ub 75cf338a
kernel32!GetEnvironmentVariableA+0xa:
75cf3377 90              nop
kernel32!BaseThreadInitThunk:
75cf3378 8bff            mov     edi,edi
75cf337a 55              push    ebp
75cf337b 8bec            mov     ebp,esp
75cf337d 85c9            test    ecx,ecx
75cf337f 0f857d2e0000    jne     kernel32!BaseThreadInitThunk+0x15 (75cf6202)
75cf3385 ff7508          push    dword ptr [ebp+8]
75cf3388 ffd2            call    edx
0:000> uf /c HelloWorld!wmain
HelloWorld!wmain (013d1430) [d:\myprojects\sourcelens\windbgintro\forbreakpoints\helloworld\helloworld
  HelloWorld!wmain+0x34 (013d1464) [d:\myprojects\sourcelens\windbgintro\forbreakpoints\helloworld\hel
    call to HelloWorld!ILT+265(?MyTestFuncYAXPADHZ) (013d110e)
  HelloWorld!wmain+0x4e (013d147e) [d:\myprojects\sourcelens\windbgintro\forbreakpoints\helloworld\hel
    call to HelloWorld!ILT+265(?MyTestFuncYAXPADHZ) (013d110e)
  HelloWorld!wmain+0x63 (013d1493) [d:\myprojects\sourcelens\windbgintro\forbreakpoints\helloworld\hel
    call to HelloWorld!ILT+315(__RTC_CheckEsp) (013d1140)

0:000>
```

Part 14: Command s or search memory

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)



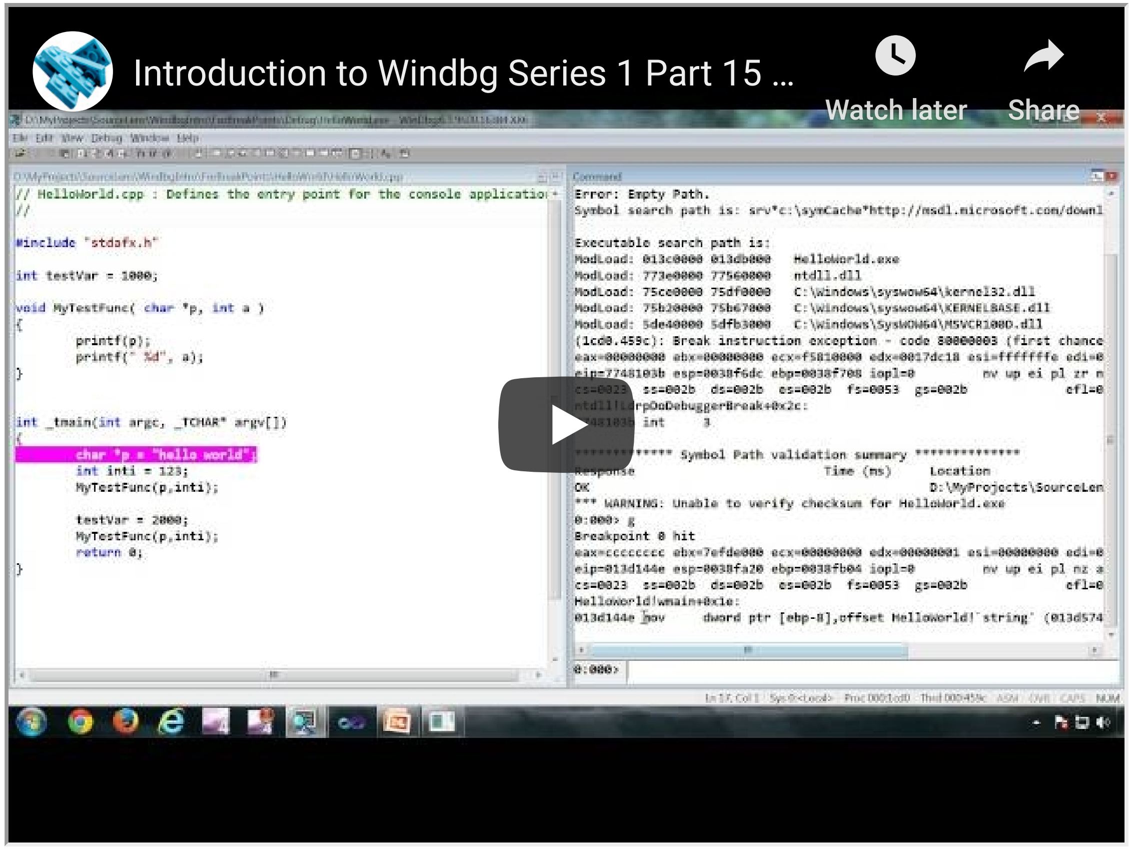
Sample

```
s 0012ff40 L20 'H' 'e' 'l' 'l' 'o'
s 0 L?7fffffff 'H' 'e' 'l' 'l' 'o'
s -a 0012ff40 L20 "Hello"
s -d 0012ff40 L20 'VUTS'
```

Part 15: Command bp for giving breakpoints

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)



Introduction to Windbg Series 1 Part 15 ...
Watch later
Share
File Edit View Debug Window Help
Command

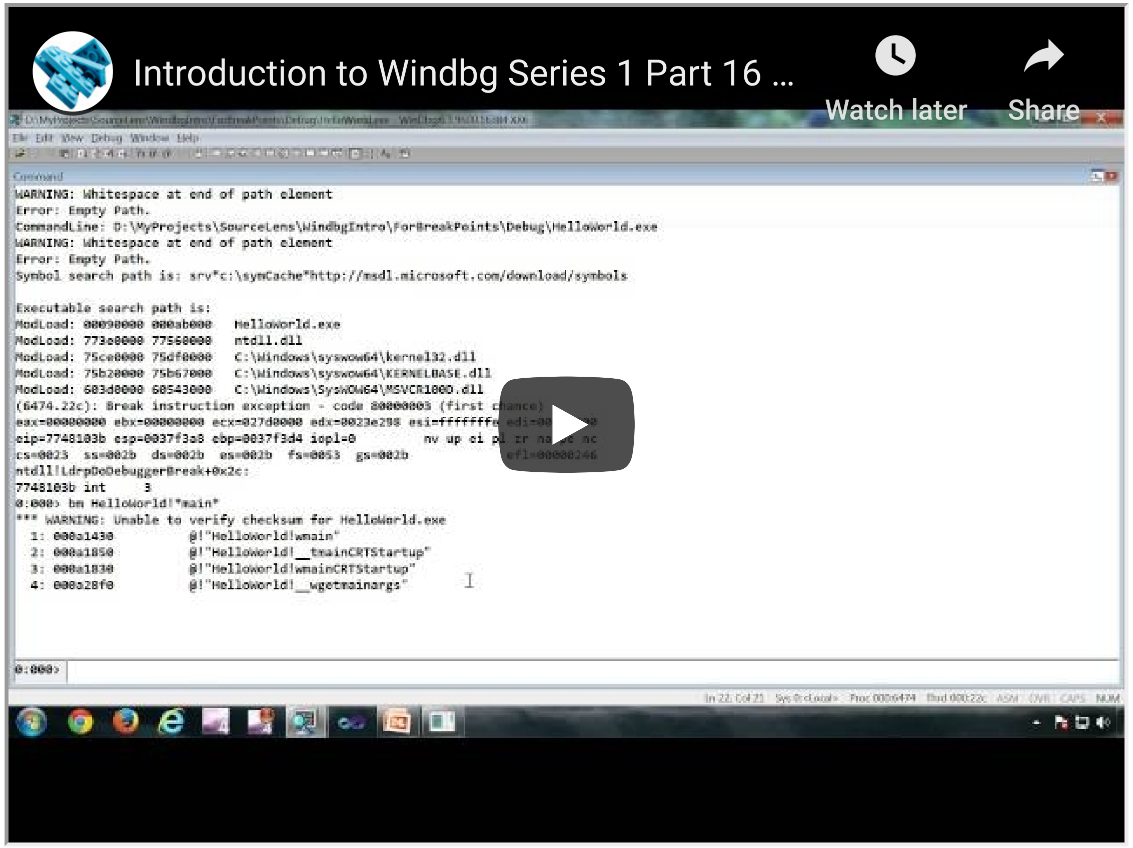
```
// HelloWorld.cpp : Defines the entry point for the console applicatio
//

#include "stdafx.h"

int testVar = 1000;

void MyTestFunc( char *p, int a )
{
        printf(p);
        printf(" %d", a);
}


int _tmain(int argc, _TCHAR* argv[])
{
        char *p = "hello world";
        int inti = 123;
        MyTestFunc(p,inti);

        testVar = 2000;
        MyTestFunc(p,inti);
        return 0;
}
```

```
Error: Empty Path.
Symbol search path is: srv*c:\symCache*http://msdl.microsoft.com/downl

Executable search path is:
ModLoad: 013c0000 013db000   HelloWorld.exe
ModLoad: 773e0000 77560000   ntdll.dll
ModLoad: 75ce0000 75df0000   C:\Windows\syswow64\kernel32.dll
ModLoad: 75b20000 75b67000   C:\Windows\syswow64\KERNELBASE.dll
ModLoad: 5de40000 5dfb3000   C:\Windows\SysWOW64\MSVCR100D.dll
(1cd0.459c): Break instruction exception - code 80000003 (first chance
eax=00000000 ebx=00000000 ecx=f5810000 edx=0017dc18 esi=fffffffe edi=0
eip=7748103b esp=0038f6dc ebp=0038f708 iopl=0         nv up ei pl zr n
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=0
ntdll!LdrpDoDebuggerBreak+0x2c:
7748103b int     3

************* Symbol Path validation summary **************
Response                         Time (ms)     Location
OK                                             D:\MyProjects\SourceLen
*** WARNING: Unable to verify checksum for HelloWorld.exe
0:000> g
Breakpoint 0 hit
eax=cccccccc ebx=7efde000 ecx=00000000 edx=00000001 esi=00000000 edi=0
eip=013d144e esp=0038fa20 ebp=0038fb04 iopl=0         nv up ei pl nz a
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=0
HelloWorld!wmain+0x1e:
013d144e mov     dword ptr [ebp-8],offset HelloWorld!`string' (013d574

0:000>
```

Part 16: Command bm for break point

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)

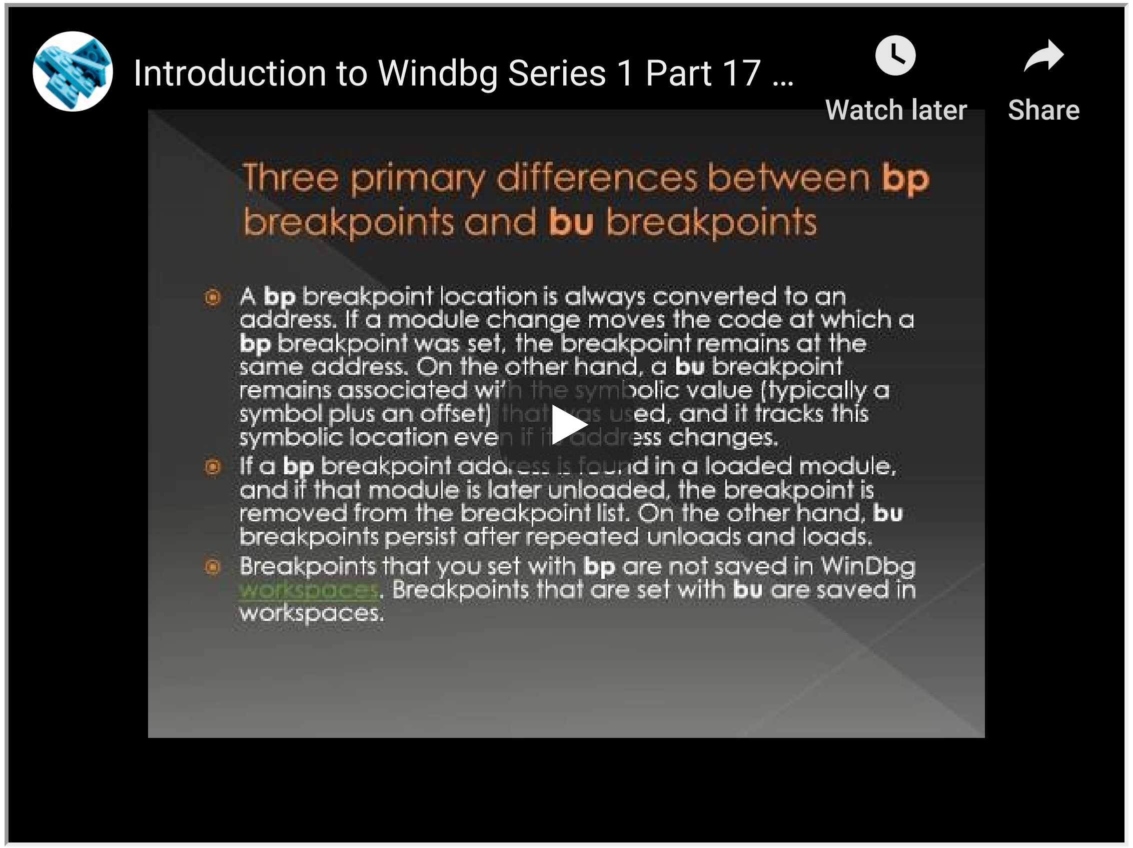

Introduction to Windbg Series 1 Part 16 ...
Watch later
Share
File Edit View Debug Window Help
Command

```
WARNING: Whitespace at end of path element
Error: Empty Path.
CommandLine: D:\MyProjects\SourceLens\WindbgIntro\ForBreakPoints\Debug\HelloWorld.exe
WARNING: Whitespace at end of path element
Error: Empty Path.
Symbol search path is: srv*c:\symCache*http://msdl.microsoft.com/download/symbols

Executable search path is:
ModLoad: 00090000 000ab000   HelloWorld.exe
ModLoad: 773e0000 77560000   ntdll.dll
ModLoad: 75ce0000 75df0000   C:\Windows\syswow64\kernel32.dll
ModLoad: 75b20000 75b67000   C:\Windows\syswow64\KERNELBASE.dll
ModLoad: 603d0000 60543000   C:\Windows\SysWOW64\MSVCR100D.dll
(6474.22c): Break instruction exception - code 80000003 (first chance)
eax=00000000 ebx=00000000 ecx=027d0000 edx=0023e298 esi=fffffffe edi=00[illegible]
eip=7748103b esp=0037f3a8 ebp=0037f3d4 iopl=0         nv up ei pl zr na pe nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000246
ntdll!LdrpDoDebuggerBreak+0x2c:
7748103b int     3
0:000> bm HelloWorld!*main*
*** WARNING: Unable to verify checksum for HelloWorld.exe
  1: 000a1430          @!"HelloWorld!wmain"
  2: 000a1850          @!"HelloWorld!__tmainCRTStartup"
  3: 000a1830          @!"HelloWorld!wmainCRTStartup"
  4: 000a28f0          @!"HelloWorld!__wgetmainargs"

0:000>
```

Part 17: Command bu or breakpoint unresolved

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)



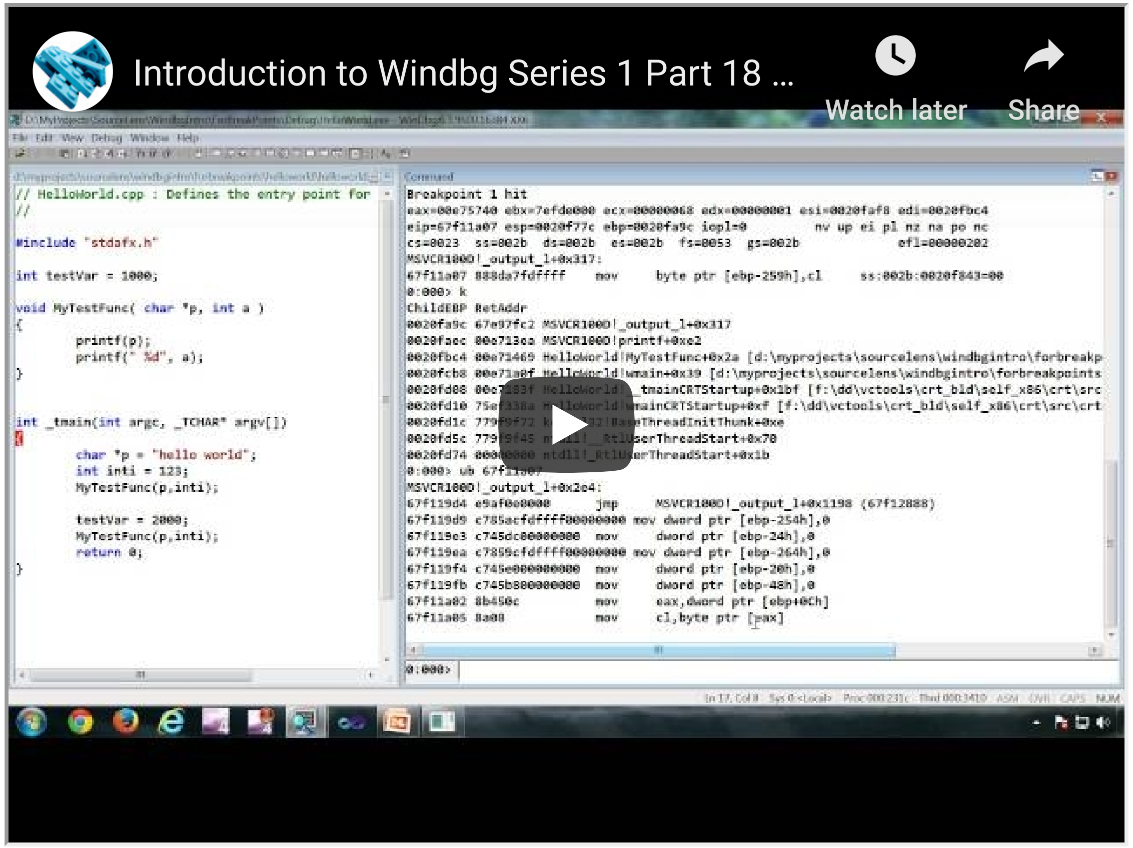
Three primary differences between **bp** breakpoints and **bu** breakpoints

- A **bp** breakpoint location is always converted to an address. If a module change moves the code at which a **bp** breakpoint was set, the breakpoint remains at the same address. On the other hand, a **bu** breakpoint remains associated with the symbolic value (typically a symbol plus an offset) that was used, and it tracks this symbolic location even if its address changes.
- If a **bp** breakpoint address is found in a loaded module, and if that module is later unloaded, the breakpoint is removed from the breakpoint list. On the other hand, **bu** breakpoints persist after repeated unloads and loads.
- Breakpoints that you set with **bp** are not saved in WinDbg workspaces. Breakpoints that are set with **bu** are saved in workspaces.

Part 18: Command ba or break on access

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)


Introduction to Windbg Series 1 Part 18 ...
Watch later
Share
File Edit View Debug Window Help
Command

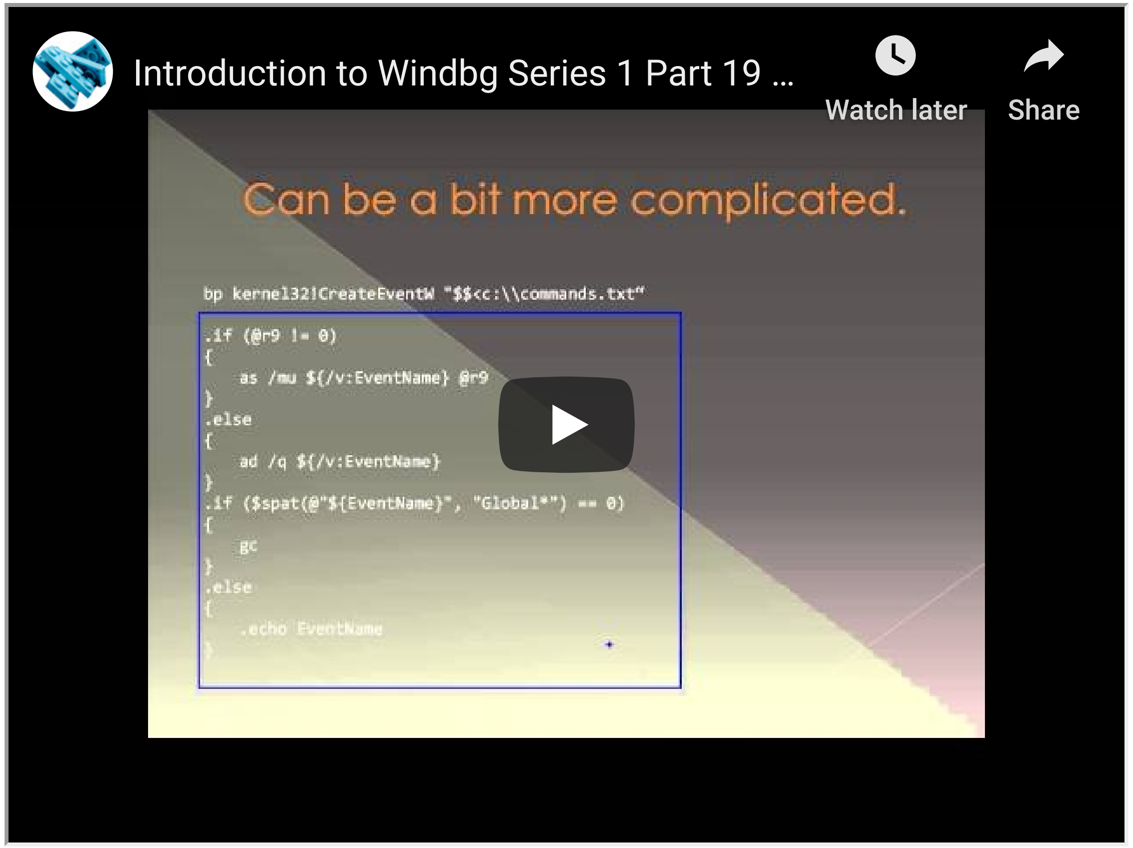
```
// HelloWorld.cpp : Defines the entry point for
//

#include "stdafx.h"

int testVar = 1000;

void MyTestFunc( char *p, int a )
{
        printf(p);
        printf(" %d", a);
}


int _tmain(int argc, _TCHAR* argv[])
{
        char *p = "hello world";
        int inti = 123;
        MyTestFunc(p,inti);

        testVar = 2000;
        MyTestFunc(p,inti);
        return 0;
}
```

```
Breakpoint 1 hit
eax=00e75740 ebx=7efde000 ecx=00000068 edx=00000001 esi=0020faf8 edi=0020fbc4
eip=67f11a07 esp=0020f77c ebp=0020fa9c iopl=0         nv up ei pl nz na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00000202
MSVCR100D!_output_l+0x317:
67f11a07 888da7fdffff    mov     byte ptr [ebp-259h],cl     ss:002b:0020f843=00
0:000> k
ChildEBP RetAddr
0020fa9c 67e97fc2 MSVCR100D!_output_l+0x317
0020faec 00e713ea MSVCR100D!printf+0xe2
0020fbc4 00e71469 HelloWorld!MyTestFunc+0x2a [d:\myprojects\sourcelens\windbgintro\forbreakp
0020fcb8 00e71a0f HelloWorld!wmain+0x39 [d:\myprojects\sourcelens\windbgintro\forbreakpoints
0020fd08 00e7183f HelloWorld!__tmainCRTStartup+0x1bf [f:\dd\vctools\crt_bld\self_x86\crt\src
0020fd10 75ef338a HelloWorld!wmainCRTStartup+0xf [f:\dd\vctools\crt_bld\self_x86\crt\src\crt
0020fd1c 779f9f72 kernel32!BaseThreadInitThunk+0xe
0020fd5c 779f9f45 ntdll!__RtlUserThreadStart+0x70
0020fd74 00000000 ntdll!_RtlUserThreadStart+0x1b
0:000> ub 67f11a07
MSVCR100D!_output_l+0x2e4:
67f119d4 e9af0e0000      jmp     MSVCR100D!_output_l+0x1198 (67f12888)
67f119d9 c785acfdffff00000000 mov dword ptr [ebp-254h],0
67f119e3 c745dc00000000  mov     dword ptr [ebp-24h],0
67f119ea c7859cfdffff00000000 mov dword ptr [ebp-264h],0
67f119f4 c745e000000000  mov     dword ptr [ebp-20h],0
67f119fb c745b800000000  mov     dword ptr [ebp-48h],0
67f11a02 8b450c          mov     eax,dword ptr [ebp+0Ch]
67f11a05 8a08            mov     cl,byte ptr [eax]

0:000>
```

Part 19: Conditional breakpoints

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)



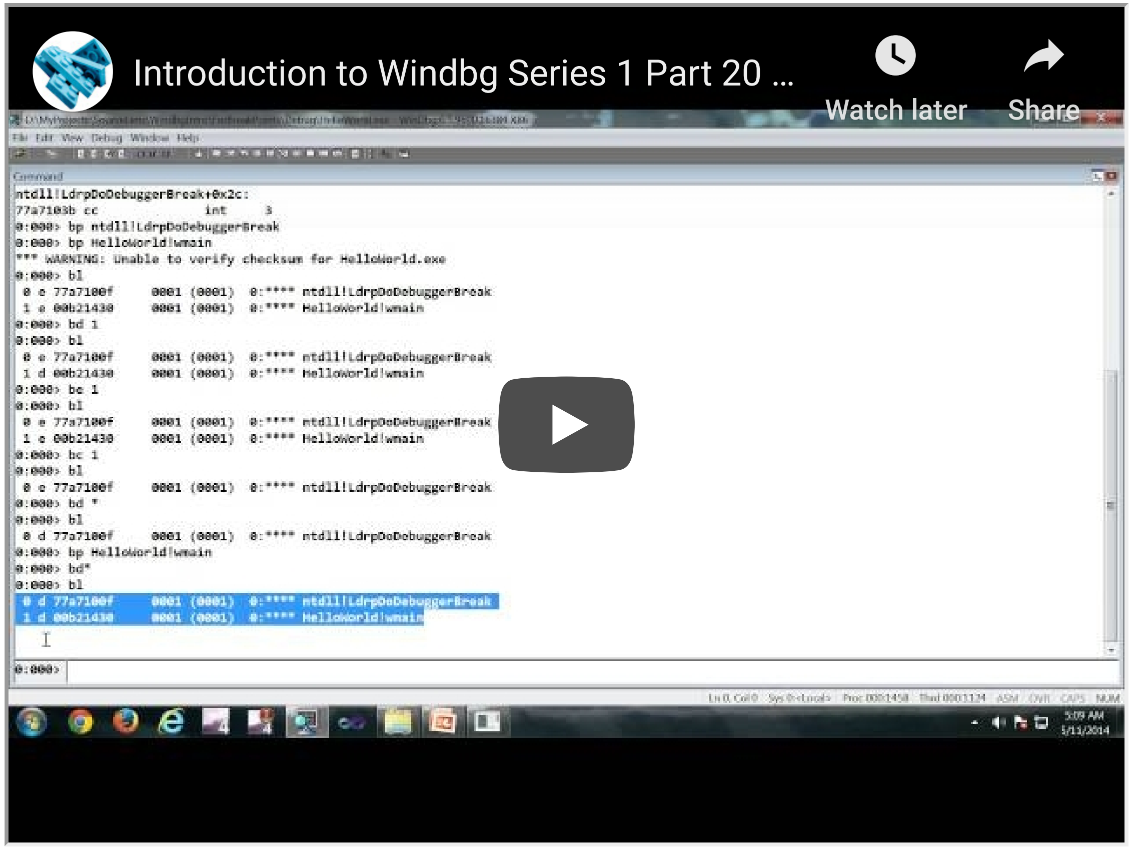
Can be a bit more complicated.

```
bp kernel32!CreateEventW "$$<c:\\commands.txt"
.if (@r9 != 0)
{
   as /mu ${/v:EventName} @r9
}
.else
{
   ad /q ${/v:EventName}
}
.if ($spat(@"${EventName}", "Global*") == 0)
{
   gc
}
.else
{
   .echo EventName
}
```

Part 20: Miscellaneous breakpoint related commands

SECURITY CHOPS

growing my chops in cybersecurity
(all opinions are my own and not the views of my employer)



Introduction to Windbg Series 1 Part 20 ...
Watch later
Share
File Edit View Debug Window Help
Command

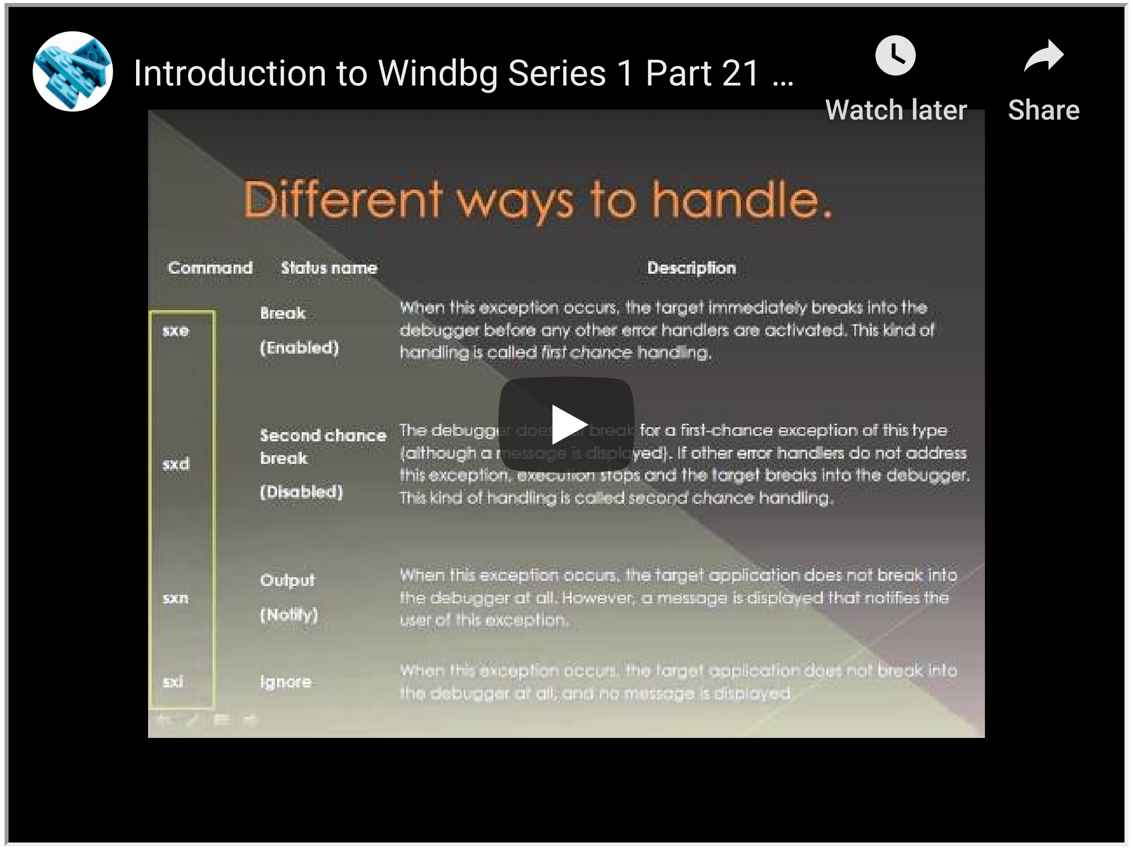
```
ntdll!LdrpDoDebuggerBreak+0x2c:
77a7103b cc              int     3
0:000> bp ntdll!LdrpDoDebuggerBreak
0:000> bp HelloWorld!wmain
*** WARNING: Unable to verify checksum for HelloWorld.exe
0:000> bl
 0 e 77a7100f     0001 (0001)  0:**** ntdll!LdrpDoDebuggerBreak
 1 e 00b21430     0001 (0001)  0:**** HelloWorld!wmain
0:000> bd 1
0:000> bl
 0 e 77a7100f     0001 (0001)  0:**** ntdll!LdrpDoDebuggerBreak
 1 d 00b21430     0001 (0001)  0:**** HelloWorld!wmain
0:000> be 1
0:000> bl
 0 e 77a7100f     0001 (0001)  0:**** ntdll!LdrpDoDebuggerBreak
 1 e 00b21430     0001 (0001)  0:**** HelloWorld!wmain
0:000> bc 1
0:000> bl
 0 e 77a7100f     0001 (0001)  0:**** ntdll!LdrpDoDebuggerBreak
0:000> bd *
0:000> bl
 0 d 77a7100f     0001 (0001)  0:**** ntdll!LdrpDoDebuggerBreak
0:000> bp HelloWorld!wmain
0:000> bd*
0:000> bl
 0 d 77a7100f     0001 (0001)  0:**** ntdll!LdrpDoDebuggerBreak
 1 d 00b21430     0001 (0001)  0:**** HelloWorld!wmain

0:000>
```

Part 21: Exceptions And Events

SECURITY CHOPS

growing my chops in cybersecurity

(all opinions are my own and not the views of my employer)



Different ways to handle.

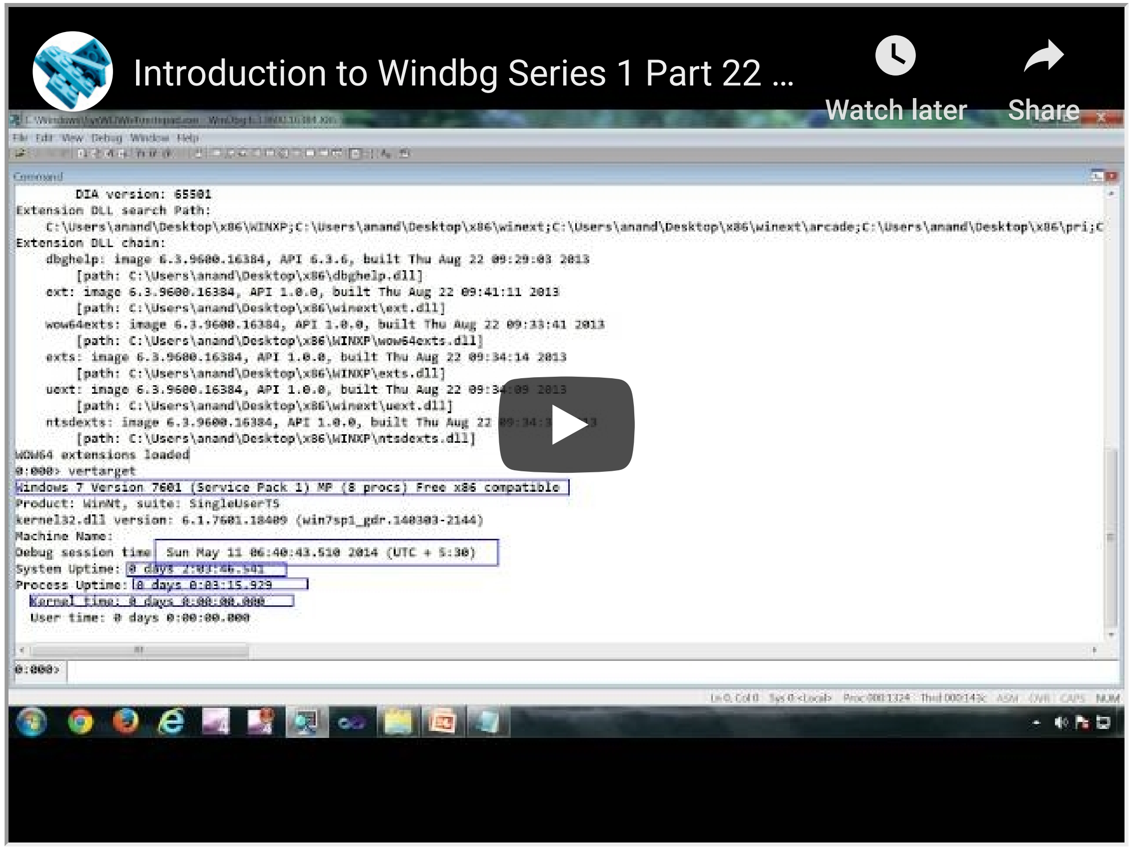
Command	Status name	Description
sxe	Break (Enabled)	When this exception occurs, the target immediately breaks into the debugger before any other error handlers are activated. This kind of handling is called *first chance handling*.
sxd	Second chance break (Disabled)	The debugger does not break for a first-chance exception of this type (although a message is displayed). If other error handlers do not address this exception, execution stops and the target breaks into the debugger. This kind of handling is called *second chance handling*.
sxn	Output (Notify)	When this exception occurs, the target application does not break into the debugger at all. However, a message is displayed that notifies the user of this exception.
sxi	Ignore	When this exception occurs, the target application does not break into the debugger at all, and no message is displayed.

Part 22: Miscellaneous Commands

SECURITY CHOPS

growing my chops in cybersecurity

(all opinions are my own and not the views of my employer)



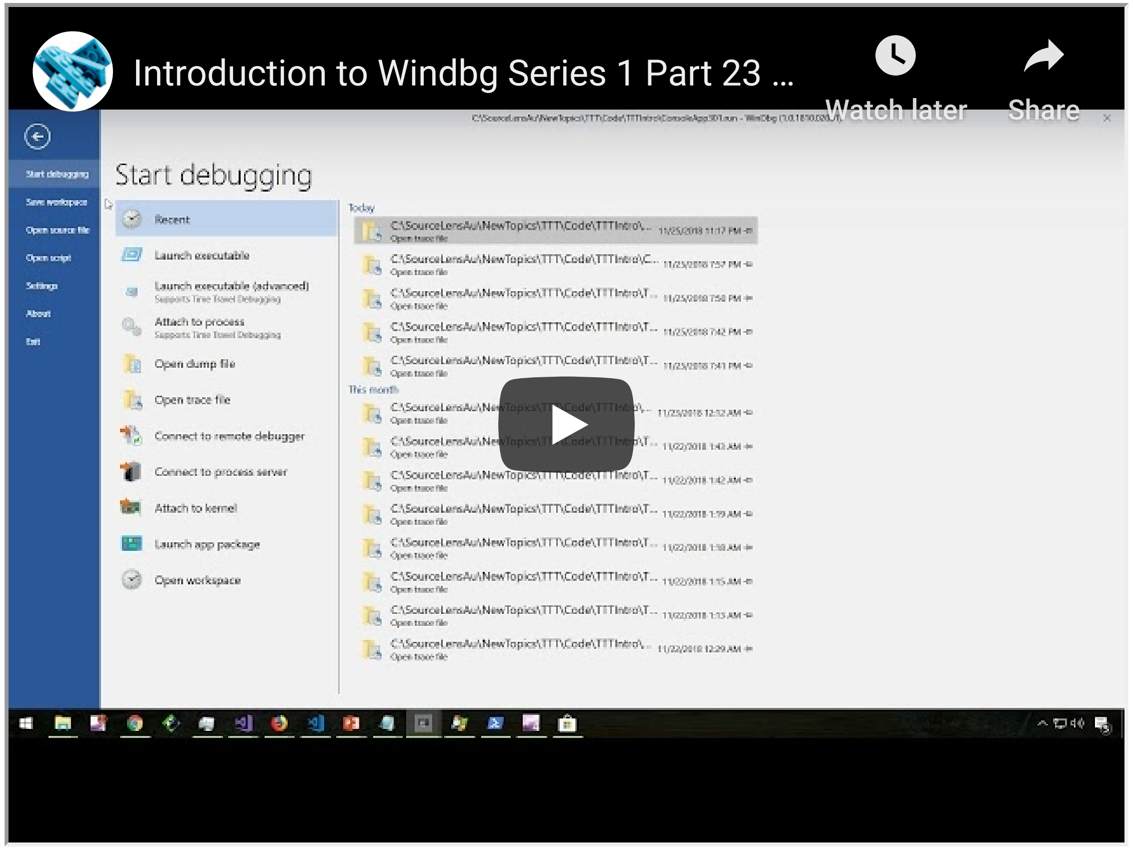
Introduction to Windbg Series 1 Part 22 ...

```
        DIA version: 65501
Extension DLL search Path:
    C:\Users\anand\Desktop\x86\WINXP;C:\Users\anand\Desktop\x86\winext;C:\Users\anand\Desktop\x86\winext\arcade;C:\Users\anand\Desktop\x86\pri;C
Extension DLL chain:
    dbghelp: image 6.3.9600.16384, API 6.3.6, built Thu Aug 22 09:29:03 2013
        [path: C:\Users\anand\Desktop\x86\dbghelp.dll]
    ext: image 6.3.9600.16384, API 1.0.0, built Thu Aug 22 09:41:11 2013
        [path: C:\Users\anand\Desktop\x86\winext\ext.dll]
    wow64exts: image 6.3.9600.16384, API 1.0.0, built Thu Aug 22 09:33:41 2013
        [path: C:\Users\anand\Desktop\x86\WINXP\wow64exts.dll]
    exts: image 6.3.9600.16384, API 1.0.0, built Thu Aug 22 09:34:14 2013
        [path: C:\Users\anand\Desktop\x86\WINXP\exts.dll]
    uext: image 6.3.9600.16384, API 1.0.0, built Thu Aug 22 09:34:09 2013
        [path: C:\Users\anand\Desktop\x86\winext\uext.dll]
    ntsdexts: image 6.3.9600.16384, API 1.0.0, built Thu Aug 22 09:34:3[illegible]13
        [path: C:\Users\anand\Desktop\x86\WINXP\ntsdexts.dll]
WOW64 extensions loaded
0:000> vertarget
Windows 7 Version 7601 (Service Pack 1) MP (8 procs) Free x86 compatible
Product: WinNt, suite: SingleUserTS
kernel32.dll version: 6.1.7601.18409 (win7sp1_gdr.140303-2144)
Machine Name:
Debug session time: Sun May 11 06:40:43.510 2014 (UTC + 5:30)
System Uptime: 0 days 2:03:46.541
Process Uptime: 0 days 0:03:15.929
  Kernel time: 0 days 0:00:00.000
  User time: 0 days 0:00:00.000

0:000>
```

Part 23: Time travellers tracing (IDNA)

SECURITY CHOPS

growing my chops in cybersecurity (all opinions are my own and not the views of my employer)



/dev/random (4) , windbg (2) , debuggers (3)

Share Post

Twitter Facebook

Jonathan Crosby

growing my chops in cybersecurity

(all opinions are my own and not the views of my employer)

SECURITY CHOPS

growing my chops in cybersecurity

(all opinions are my own and not the views of my employer)