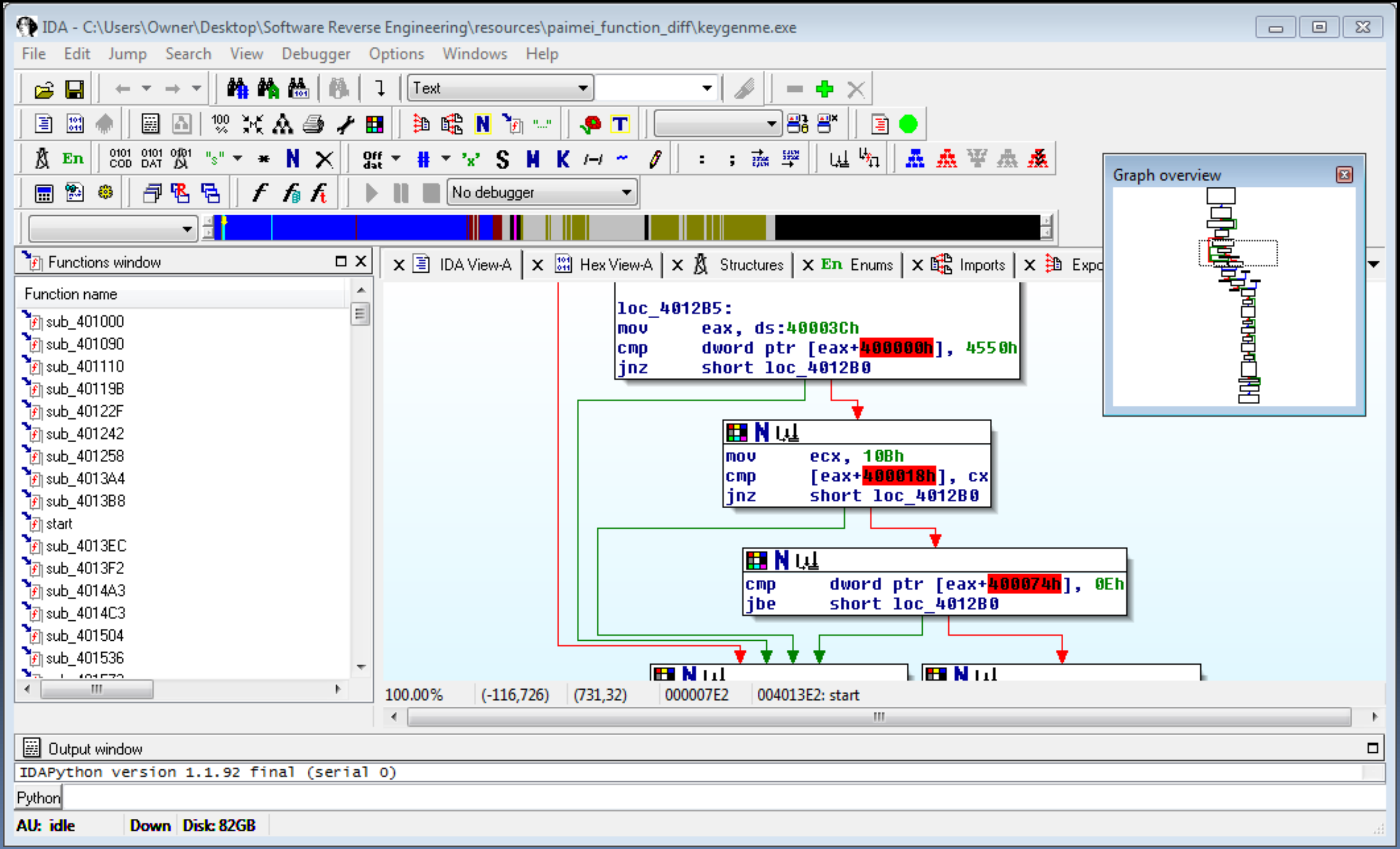
REVERSE ENGINEERING MACHINE CODE: PART 3

Code Visualization


IDA - C:\Users\Owner\Desktop\Software Reverse Engineering\resources\paimei_function_diff\keygenme.exe
File Edit Jump Search View Debugger Options Windows Help
Text
No debugger
Functions window
Function name
IDA View-A
Hex View-A
Structures
Enums
Imports
Graph overview
Output window
Python

```
sub_401000
sub_401090
sub_401110
sub_40119B
sub_40122F
sub_401242
sub_401258
sub_4013A4
sub_4013B8
start
sub_4013EC
sub_4013F2
sub_4014A3
sub_4014C3
sub_401504
sub_401536
```

```
loc_4012B5:
mov     eax, ds:40003Ch
cmp     dword ptr [eax+400000h], 4550h
jnz     short loc_4012B0
```

```
mov     ecx, 10Bh
cmp     [eax+400018h], cx
jnz     short loc_4012B0
```

```
cmp     dword ptr [eax+400074h], 0Eh
jbe     short loc_4012B0
```

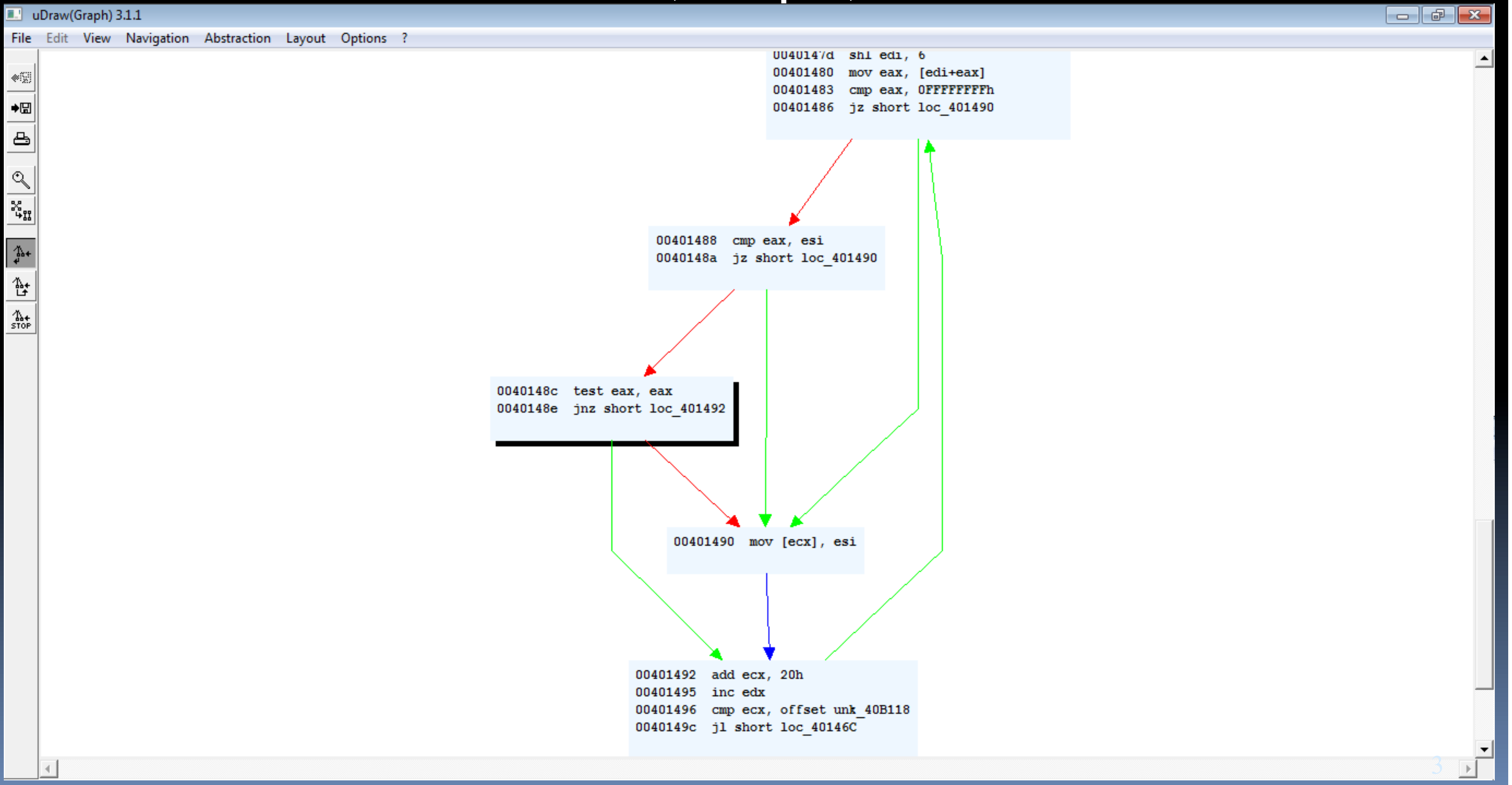
```
100.00%   (-116,726)   (731,32)   000007E2   004013E2: start
```

```
IDAPython version 1.1.92 final (serial 0)
```

AU: idle   Down   Disk: 82GB

Code Visualization

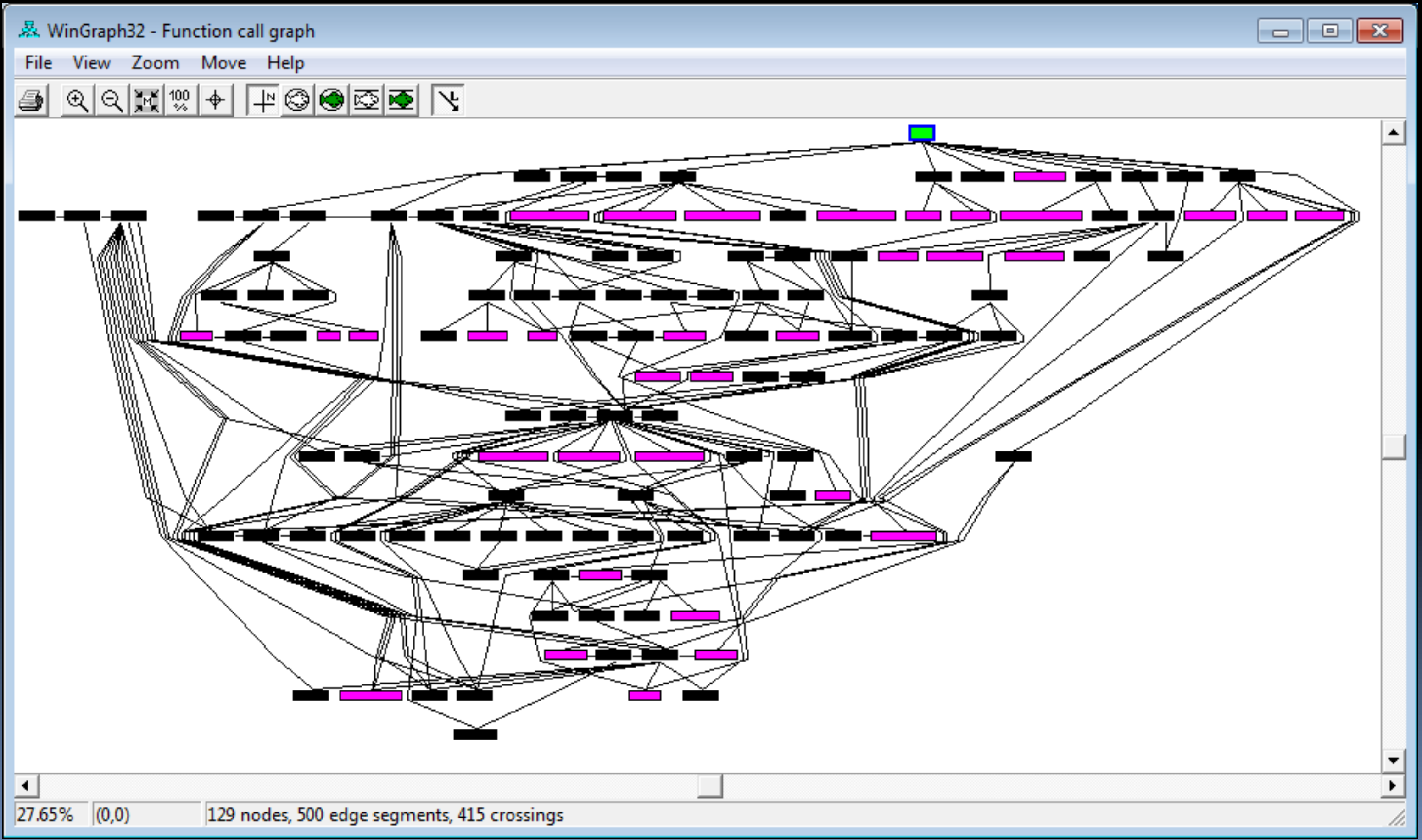
- PaiMei and uDraw(Graph)



```
0040147d  shl edi, 6
00401480  mov eax, [edi+eax]
00401483  cmp eax, 0FFFFFFFFh
00401486  jz short loc_401490

00401488  cmp eax, esi
0040148a  jz short loc_401490

0040148c  test eax, eax
0040148e  jnz short loc_401492

00401490  mov [ecx], esi

00401492  add ecx, 20h
00401495  inc edx
00401496  cmp ecx, offset unk_40B118
0040149c  jl short loc_40146C
```

Code Visualization

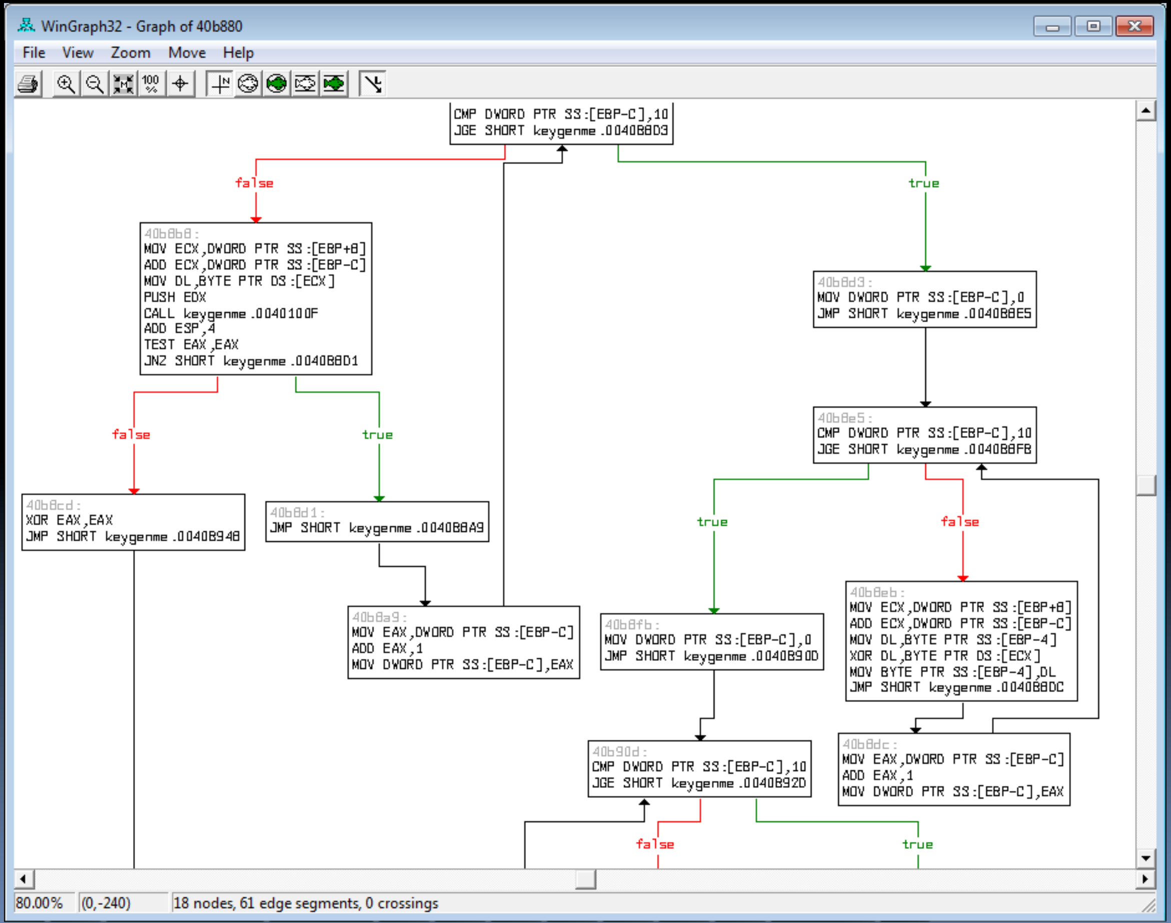
- OllyFlow Plugin: Function graphs



129 nodes, 500 edge segments, 415 crossings

Code Visualization

- OllyFlow Plugin: Flow graphs



```
CMP DWORD PTR SS:[EBP-C],10
JGE SHORT keygenme.0040B8D3
```

```
40b8b8:
MOV ECX,DWORD PTR SS:[EBP+8]
ADD ECX,DWORD PTR SS:[EBP-C]
MOV DL,BYTE PTR DS:[ECX]
PUSH EDX
CALL keygenme.0040100F
ADD ESP,4
TEST EAX,EAX
JNZ SHORT keygenme.0040B8D1
```

```
40b8cd:
XOR EAX,EAX
JMP SHORT keygenme.0040B948
```

```
40b8d1:
JMP SHORT keygenme.0040B8A9
```

```
40b8a9:
MOV EAX,DWORD PTR SS:[EBP-C]
ADD EAX,1
MOV DWORD PTR SS:[EBP-C],EAX
```

```
40b8d3:
MOV DWORD PTR SS:[EBP-C],0
JMP SHORT keygenme.0040B8E5
```

```
40b8e5:
CMP DWORD PTR SS:[EBP-C],10
JGE SHORT keygenme.0040B8FB
```

```
40b8fb:
MOV DWORD PTR SS:[EBP-C],0
JMP SHORT keygenme.0040B90D
```

```
40b8eb:
MOV ECX,DWORD PTR SS:[EBP+8]
ADD ECX,DWORD PTR SS:[EBP-C]
MOV DL,BYTE PTR SS:[EBP-4]
XOR DL,BYTE PTR DS:[ECX]
MOV BYTE PTR SS:[EBP-4],DL
JMP SHORT keygenme.0040B8DC
```

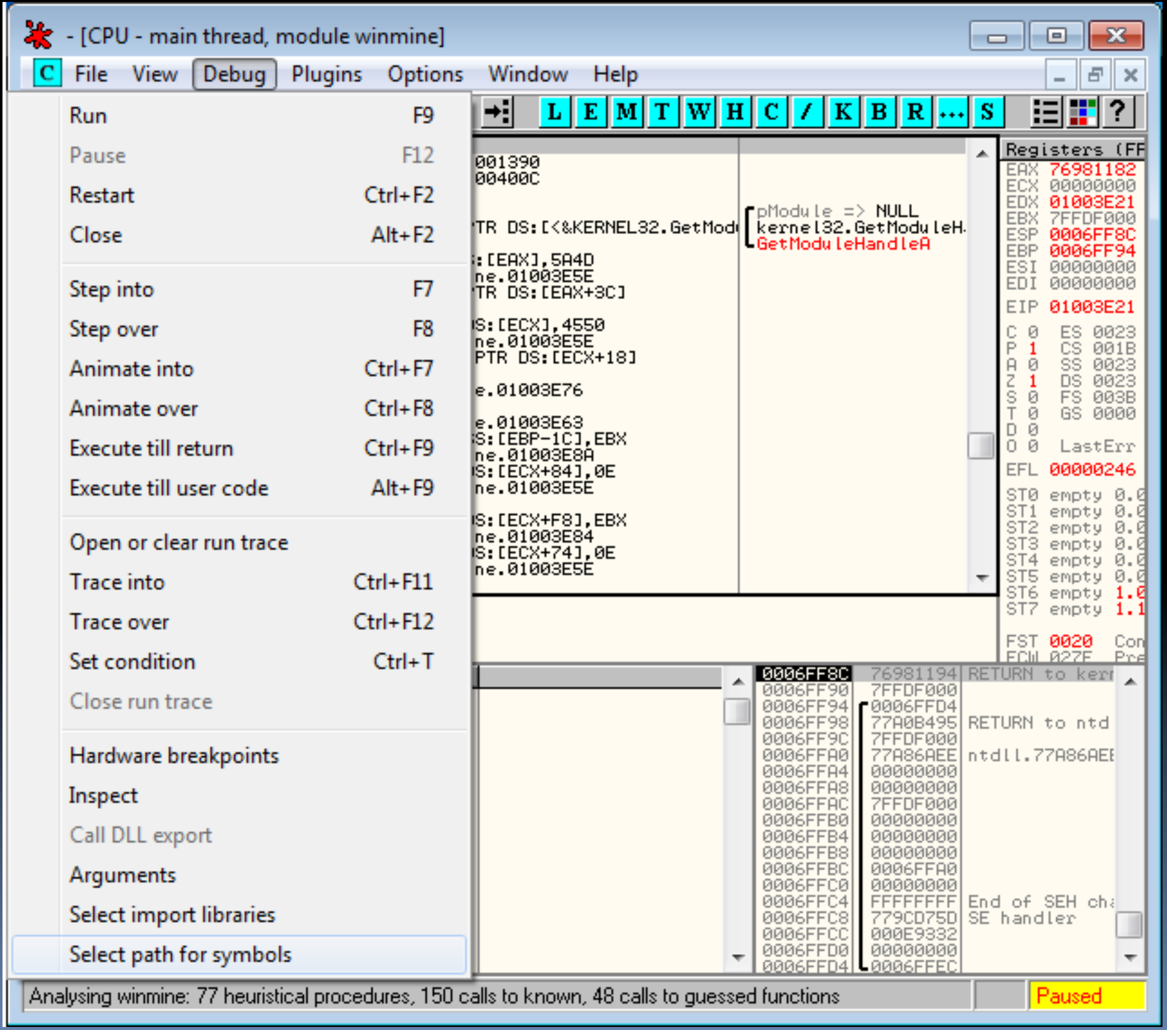
```
40b90d:
CMP DWORD PTR SS:[EBP-C],10
JGE SHORT keygenme.0040B92D
```

```
40b8dc:
MOV EAX,DWORD PTR SS:[EBP-C]
ADD EAX,1
MOV DWORD PTR SS:[EBP-C],EAX
```

Microsoft Symbols

- Debug Symbols
  - Windows kernel symbols available
    - For most MS executables
  - Windows debug symbols available at:
    - *http://www.microsoft.com/whdc/devtools/debugging/symbolpkg.mspx#f*

Configuring OllyDbg to Use Symbols



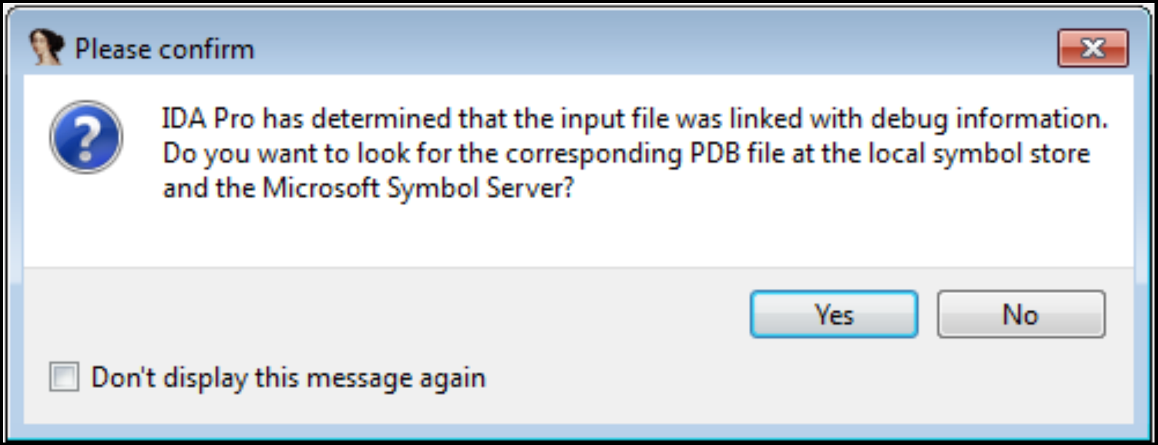
```
001390
00400C

TR DS:[<&KERNEL32.GetModu    pModule => NULL
                             kernel32.GetModuleH
                             GetModuleHandleA
:[EAX],5A4D
ne.01003E5E
TR DS:[EAX+3C]

S:[ECX],4550
ne.01003E5E
PTR DS:[ECX+18]

e.01003E76

e.01003E63
S:[EBP-1C],EBX
ne.01003E8A
S:[ECX+84],0E
ne.01003E5E

S:[ECX+F8],EBX
ne.01003E84
S:[ECX+74],0E
ne.01003E5E
```

```
Registers (FF
EAX 76981182
ECX 00000000
EDX 01003E21
EBX 7FFDF000
ESP 0006FF8C
EBP 0006FF94
ESI 00000000
EDI 00000000

EIP 01003E21

C 0  ES 0023
P 1  CS 001B
A 0  SS 0023
Z 1  DS 0023
S 0  FS 003B
T 0  GS 0000
D 0
O 0  LastErr

EFL 00000246

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 1.0
ST7 empty 1.1

FST 0020  Con
FCW 027F  Pre
```

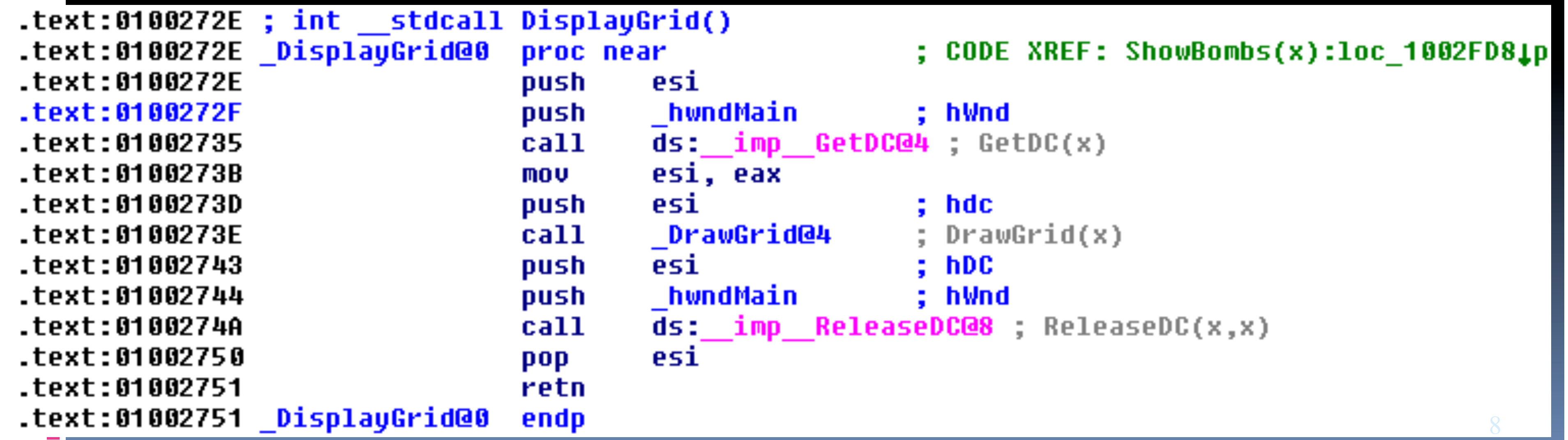
```
0006FF8C  76981194  RETURN to kern
0006FF90  7FFDF000
0006FF94  0006FFD4
0006FF98  77A0B495  RETURN to ntd
0006FF9C  7FFDF000
0006FFA0  77A86AEE  ntdll.77A86AEE
0006FFA4  00000000
0006FFA8  00000000
0006FFAC  7FFDF000
0006FFB0  00000000
0006FFB4  00000000
0006FFB8  00000000
0006FFBC  0006FFA0
0006FFC0  00000000
0006FFC4  FFFFFFFF  End of SEH cha
0006FFC8  779CD75D  SE handler
0006FFCC  000E9332
0006FFD0  00000000
0006FFD4  0006FFEC
```

Analysing winmine: 77 heuristical procedures, 150 calls to known, 48 calls to guessed functions

Symbols in IDA


Please confirm

IDA Pro has determined that the input file was linked with debug information. Do you want to look for the corresponding PDB file at the local symbol store and the Microsoft Symbol Server?


```
.text:0100272E ; int __stdcall DisplayGrid()
.text:0100272E _DisplayGrid@0  proc near               ; CODE XREF: ShowBombs(x):loc_1002FD8↓p
.text:0100272E                 push    esi
.text:0100272F                 push    _hwndMain       ; hWnd
.text:01002735                 call    ds:__imp__GetDC@4 ; GetDC(x)
.text:0100273B                 mov     esi, eax
.text:0100273D                 push    esi             ; hdc
.text:0100273E                 call    _DrawGrid@4     ; DrawGrid(x)
.text:01002743                 push    esi             ; hDC
.text:01002744                 push    _hwndMain       ; hWnd
.text:0100274A                 call    ds:__imp__ReleaseDC@8 ; ReleaseDC(x,x)
.text:01002750                 pop     esi
.text:01002751                 retn
.text:01002751 _DisplayGrid@0  endp
```

Windows Kernel Debugging

- Install Windows Symbols
  - Specifically, for the target kernel
- Install Windows SDK
  - Make sure to install “Debugging Tools”
    - This includes WinDBG

Windows Kernel Debugging

- Crash Dump Analysis
  - Windows ".dmp" file
    - Snapshot of the kernel state and physical memory
    - Windows can be configured to create a crash dump upon blue screening
    - WinDBG handles crash dump analysis

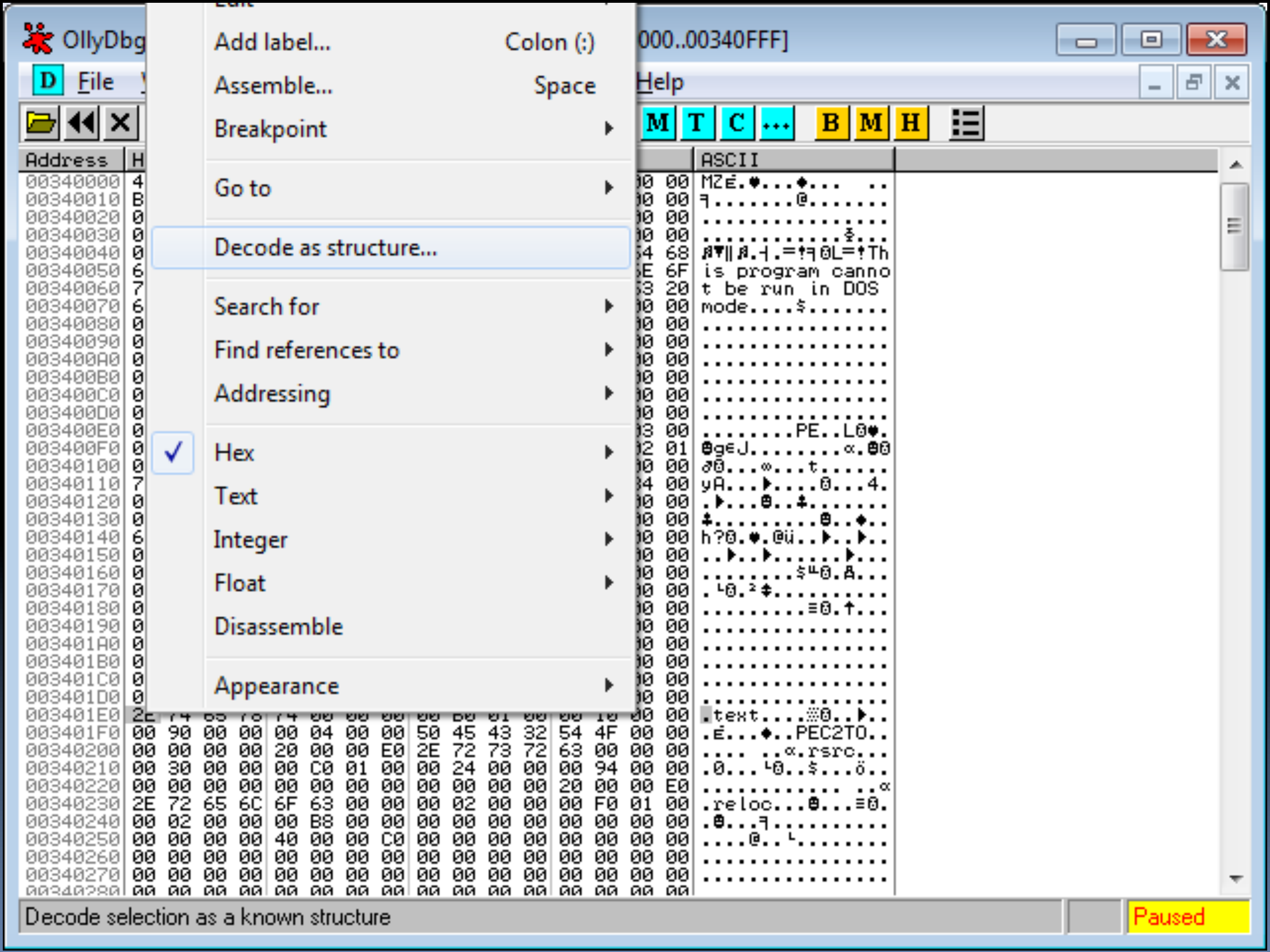
Remote Debugging

- Remote Debugging
  - Most good debuggers have a remote monitor that they can connect to
  - Some virtual machine programs incorporate remote debugging for kernel debugging

Decoding Structures

- Structures
  - Good debuggers/disassemblers will allow the user to define structures
    - Structure decoding is implemented in OllyDbg 2.0
    - Currently, user-defined structures are not

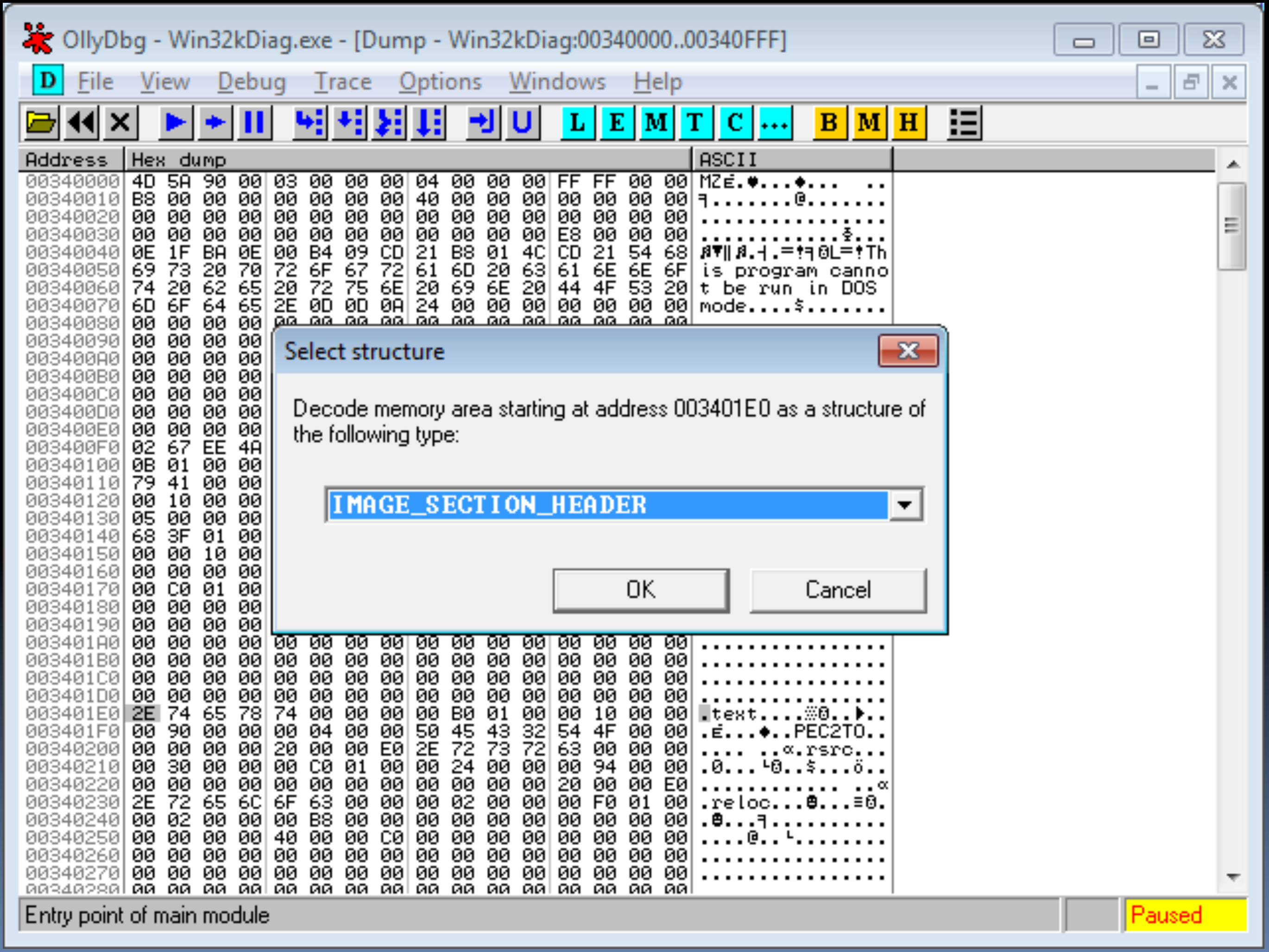
Decoding Structures



```
Address
003401F0 00 90 00 00 00 04 00 00 50 45 43 32 54 4F 00 00 .É...♦..PEC2TO..
00340200 00 00 00 00 20 00 00 E0 2E 72 73 72 63 00 00 00 ....  ..α.rsrc...
00340210 00 30 00 00 00 C0 01 00 00 24 00 00 00 94 00 00 .0...└0..$...ö..
00340220 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 E0 ............  ..α
00340230 2E 72 65 6C 6F 63 00 00 00 02 00 00 00 F0 01 00 .reloc...☻...≡0.
00340240 00 02 00 00 00 B8 00 00 00 00 00 00 00 00 00 00 .☻...╕..........
00340250 00 00 00 00 40 00 00 C0 00 00 00 00 00 00 00 00 ....@..└........
00340260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00340270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

Decode selection as a known structure

Decoding Structures


OllyDbg - Win32kDiag.exe - [Dump - Win32kDiag:00340000..00340FFF]
D File View Debug Trace Options Windows Help
L E M T C ... B M H

```
Address  Hex dump                                         ASCII
00340000 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZÉ.♥...♦...  ..
00340010 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ╕.......@.......
00340020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00340030 00 00 00 00 00 00 00 00 00 00 00 00 E8 00 00 00  ............Φ...
00340040 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ♫▼║♫.┤.═!╕☺L═!Th
00340050 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00340060 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
00340070 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00340080 00 00 00 00
00340090 00 00 00 00
003400A0 00 00 00 00
003400B0 00 00 00 00
003400C0 00 00 00 00
003400D0 00 00 00 00
003400E0 00 00 00 00
003400F0 02 67 EE 4A
00340100 0B 01 00 00
00340110 79 41 00 00
00340120 00 10 00 00
00340130 05 00 00 00
00340140 68 3F 01 00
00340150 00 00 10 00
00340160 00 00 00 00
00340170 00 C0 01 00
00340180 00 00 00 00
00340190 00 00 00 00
003401A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003401B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003401C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003401D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003401E0 2E 74 65 78 74 00 00 00 00 B0 01 00 00 10 00 00  .text....░☺..►..
003401F0 00 90 00 00 00 04 00 00 50 45 43 32 54 4F 00 00  .É...♦..PEC2TO..
00340200 00 00 00 00 20 00 00 E0 2E 72 73 72 63 00 00 00  .... ..α.rsrc...
00340210 00 30 00 00 00 C0 01 00 00 24 00 00 00 94 00 00  .0...└☺..$...ö..
00340220 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 E0  ............ ..α
00340230 2E 72 65 6C 6F 63 00 00 00 02 00 00 00 F0 01 00  .reloc...☻...≡☺.
00340240 00 02 00 00 00 B8 00 00 00 00 00 00 00 00 00 00  .☻...╕..........
00340250 00 00 00 00 40 00 00 C0 00 00 00 00 00 00 00 00  ....@..└........
00340260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00340270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00340280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Select structure

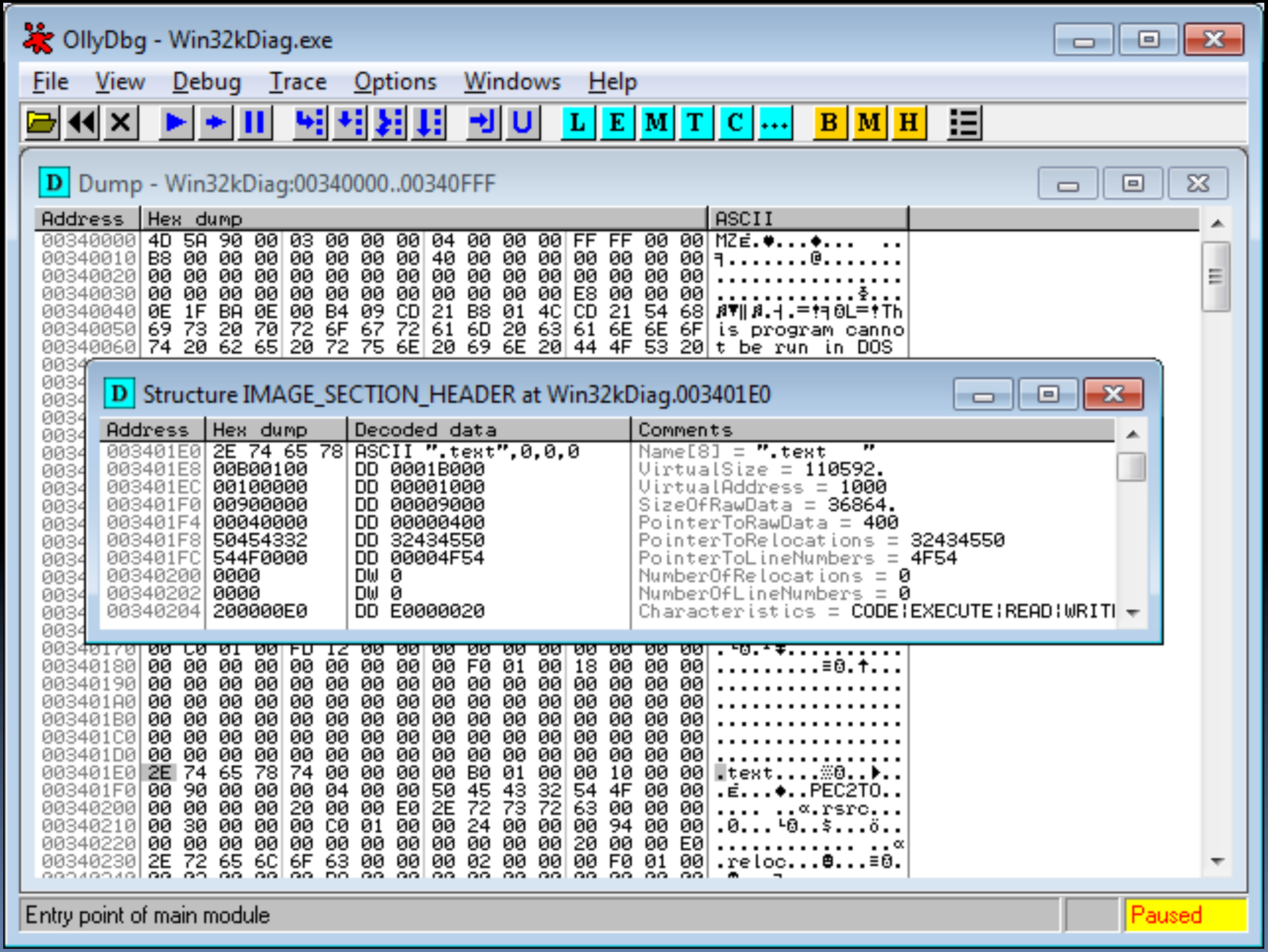
Decode memory area starting at address 003401E0 as a structure of the following type:

IMAGE_SECTION_HEADER

OK Cancel

Entry point of main module | Paused

Decoding Structures


OllyDbg - Win32kDiag.exe
File View Debug Trace Options Windows Help
L E M T C ... B M H
Dump - Win32kDiag:00340000..00340FFF
Structure IMAGE_SECTION_HEADER at Win32kDiag.003401E0
Entry point of main module
Paused

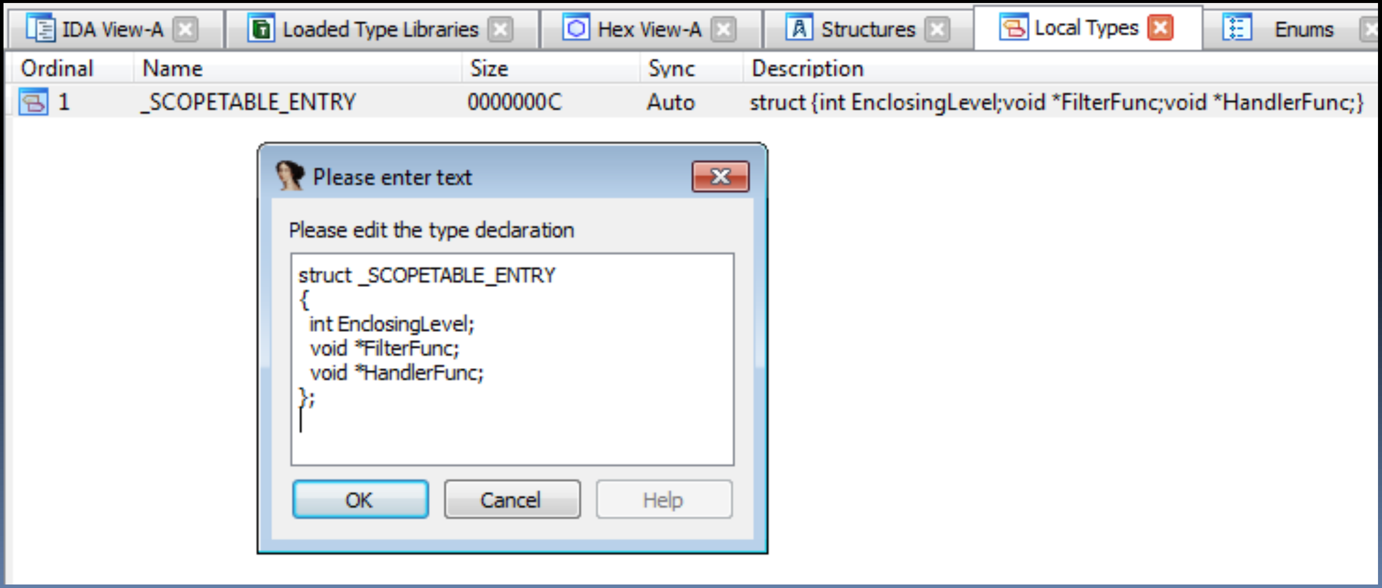
```
Address  Hex dump                                         ASCII
00340000 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 MZÉ.♥...♦...  ..
00340010 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ╕.......@.......
00340020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00340030 00 00 00 00 00 00 00 00 00 00 00 00 E8 00 00 00 ............Φ...
00340040 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 ♫▼║♫.┤.=!╕☺L=!Th
00340050 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F is program canno
00340060 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 t be run in DOS
```

```
Address  Hex dump    Decoded data           Comments
003401E0 2E 74 65 78 ASCII ".text",0,0,0   Name[8] = ".text   "
003401E8 00B00100    DD 0001B000            VirtualSize = 110592.
003401EC 00100000    DD 00001000            VirtualAddress = 1000
003401F0 00900000    DD 00009000            SizeOfRawData = 36864.
003401F4 00040000    DD 00000400            PointerToRawData = 400
003401F8 50454332    DD 32434550            PointerToRelocations = 32434550
003401FC 544F0000    DD 00004F54            PointerToLineNumbers = 4F54
00340200 0000        DW 0                   NumberOfRelocations = 0
00340202 0000        DW 0                   NumberOfLineNumbers = 0
00340204 200000E0    DD E0000020            Characteristics = CODE|EXECUTE|READ|WRITI
```

```
00340170 00 C0 01 00 FD 12 00 00 00 00 00 00 00 00 00 00 .└☺.²↕..........
00340180 00 00 00 00 00 00 00 00 00 F0 01 00 18 00 00 00 .........≡☺.↑...
00340190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
003401A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
003401B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
003401C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
003401D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
003401E0 2E 74 65 78 74 00 00 00 00 B0 01 00 00 10 00 00 .text....░☺..►..
003401F0 00 90 00 00 00 04 00 00 50 45 43 32 54 4F 00 00 .É...♦..PEC2TO..
00340200 00 00 00 00 20 00 00 E0 2E 72 73 72 63 00 00 00 .... ..α.rsrc...
00340210 00 30 00 00 00 C0 01 00 00 24 00 00 00 94 00 00 .0...└☺..$...ö..
00340220 00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 E0 ............ ..α
00340230 2E 72 65 6C 6F 63 00 00 00 02 00 00 00 F0 01 00 .reloc...☻...≡☺.
```

IDA Structures

- Assembler Structures
  - Structures window
- C Structures
  - Local types window



Ordinal	Name	Size	Sync	Description
1	_SCOPETABLE_ENTRY	0000000C	Auto	struct {int EnclosingLevel;void *FilterFunc;void *HandlerFunc;}

Please edit the type declaration

```
struct _SCOPETABLE_ENTRY
{
 int EnclosingLevel;
 void *FilterFunc;
 void *HandlerFunc;
};
```

Function Hooking

- Hooking
  - Create user-defined events upon:
    - Function calls
    - System messages
    - IO events
    - …
  - `SetWindowsHookEx()`
    - Install a hook
  - `UnhookWindowsHookEx()`
    - Uninstall a hook
  - Example: http://msdn.microsoft.com/en-us/library/windows/desktop/ms632589(v=vs.85).aspx

Differential Reverse Engineering

- Binary Diffing
- Code Coverage Diffing
- Others
  - Memory diffing

Binary Diffing

- Binary Diffing
  - Compare two similar binary executables
  - Useful for reverse engineering updates and version changes
  - Implemented in PaiMei
    - PAIMEIdiff

Code Coverage Diffing

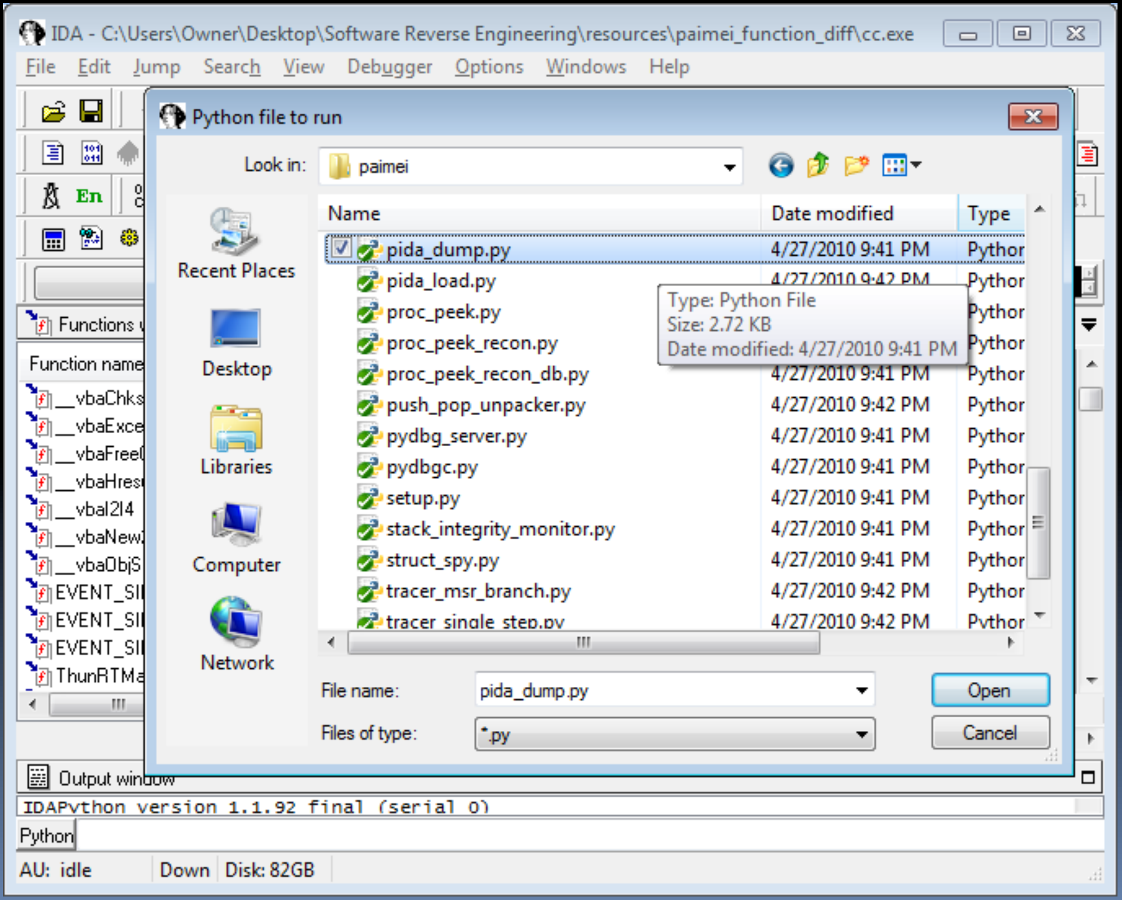
- Code Coverage Diffing
  - Typically, a reverse engineer is interested in only a few functions
  - Run 1: Profile program by running it and activating all features you aren't interested in
  - Run 2: Profile program by running just the functionality you wish to locate
  - Perform a diff on which functions were called between the two runs
  - Great for reverse engineering large or GUI programs

Code Coverage Diffing

- Demo!
  - Let's find the function in Notepad++ that invokes the "About" information
    - 3251 functions / 28827 basic blocks
    - 1 or 2 functions are dedicated to the "About" window

Code Coverage Diffing

- PaiMei's `pida_dump.py` in `idapython`


IDA - C:\Users\Owner\Desktop\Software Reverse Engineering\resources\paimei_function_diff\cc.exe
File Edit Jump Search View Debugger Options Windows Help
Python file to run
Look in: paimei
Recent Places
Desktop
Libraries
Computer
Network
Open
Cancel
Functions
Output window
Python

Name	Date modified	Type
pida_dump.py	4/27/2010 9:41 PM	Pythor
pida_load.py	4/27/2010 9:42 PM	Pythor
proc_peek.py		Pythor
proc_peek_recon.py		Pythor
proc_peek_recon_db.py	4/27/2010 9:41 PM	Pythor
push_pop_unpacker.py	4/27/2010 9:42 PM	Pythor
pydbg_server.py	4/27/2010 9:41 PM	Pythor
pydbgc.py	4/27/2010 9:41 PM	Pythor
setup.py	4/27/2010 9:41 PM	Pythor
stack_integrity_monitor.py	4/27/2010 9:41 PM	Pythor
struct_spy.py	4/27/2010 9:41 PM	Pythor
tracer_msr_branch.py	4/27/2010 9:42 PM	Pythor
tracer_single_step.py	4/27/2010 9:42 PM	Pythor

Type: Python File
Size: 2.72 KB
Date modified: 4/27/2010 9:41 PM

File name: pida_dump.py
Files of type: *.py

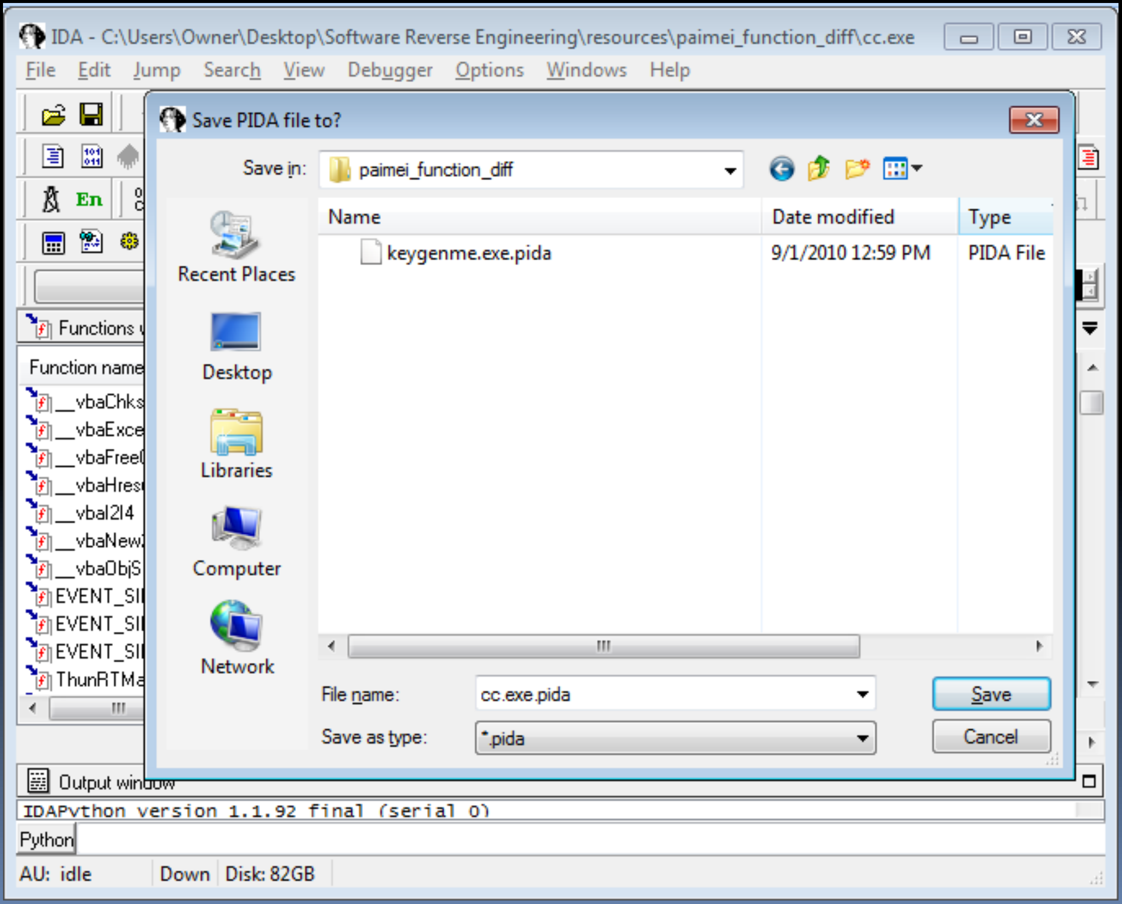
Function name
__vbaChks
__vbaExce
__vbaFree(
__vbaHres
__vbaI2I4
__vbaNew
__vbaObjS
EVENT_SII
EVENT_SII
EVENT_SII
ThunRTMa

```
IDAPython version 1.1.92 final (serial 0)
```

AU: idle | Down | Disk: 82GB

Code Coverage Diffing

- Generate a .pida file after auto-analysis



Save in: paimei_function_diff

Name	Date modified	Type
keygenme.exe.pida	9/1/2010 12:59 PM	PIDA File

File name: cc.exe.pida
Save as type: *.pida

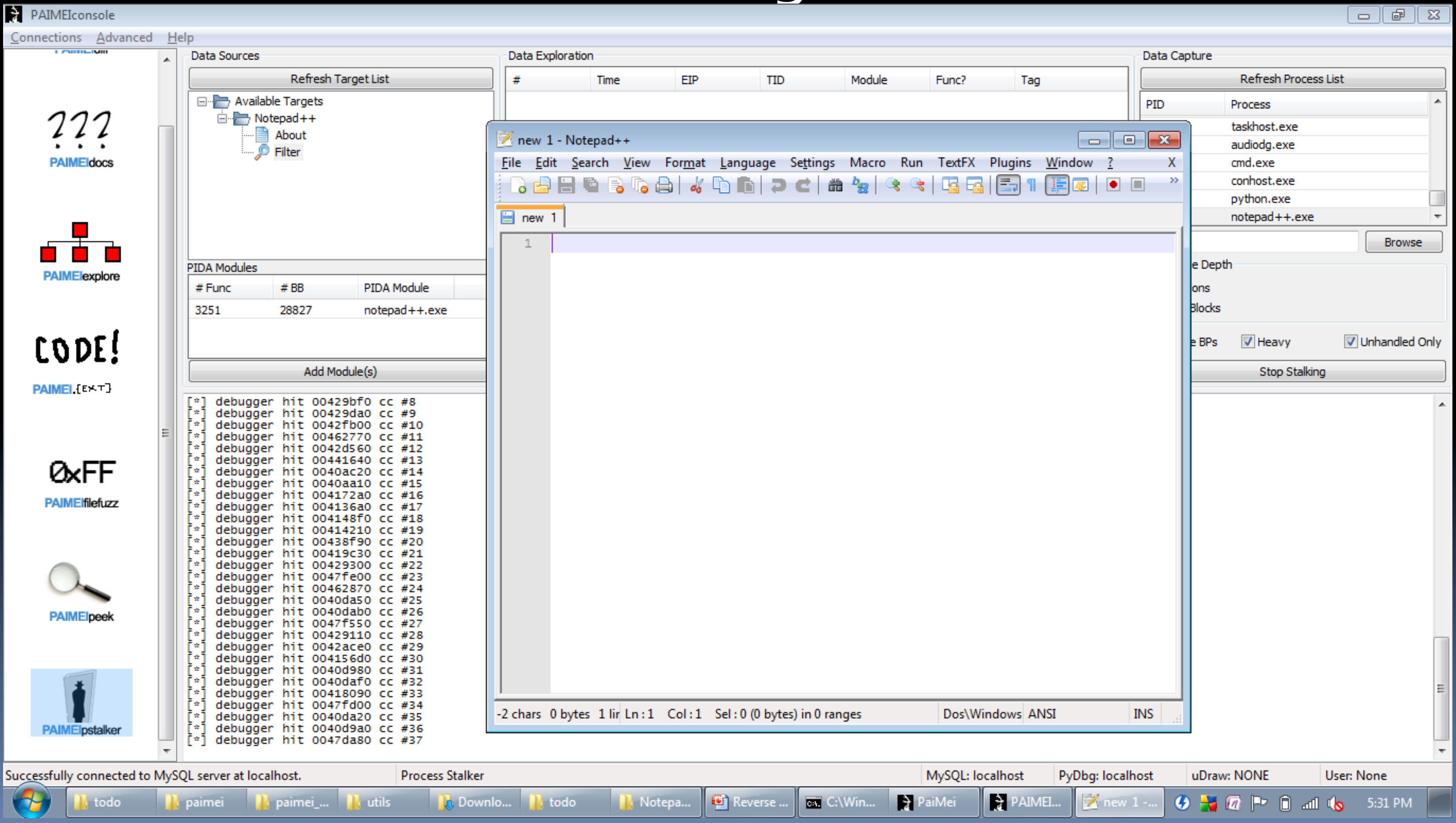
```
IDAPython version 1.1.92 final (serial 0)
```

Code Coverage Diffing

- Create Two Tags
  - “Filter”
    - We will run everything we don’t want
  - “About”
    - We will run just what we want
    - Of course, other GUI functions will be run but those should have occurred in our “Filter” tag

Code Coverage Diffing

- Use “Filter” for stalking



# Func	# BB	PIDA Module
3251	28827	notepad++.exe

#	Time	EIP	TID	Module	Func?	Tag

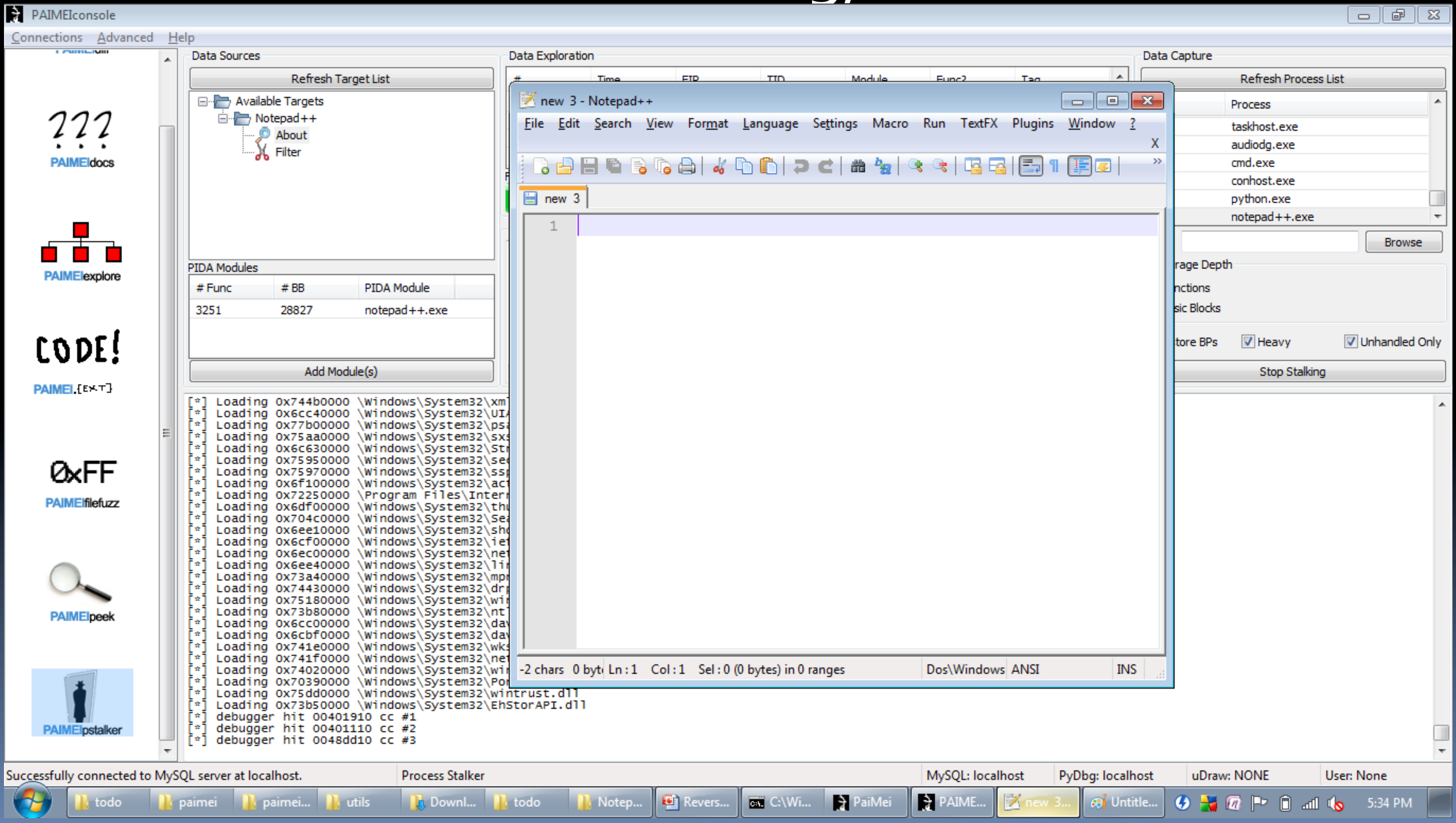
PID	Process
	taskhost.exe
	audiodg.exe
	cmd.exe
	conhost.exe
	python.exe
	notepad++.exe

```
[*] debugger hit 00429bf0 cc #8
[*] debugger hit 00429da0 cc #9
[*] debugger hit 0042fb00 cc #10
[*] debugger hit 00462770 cc #11
[*] debugger hit 0042d560 cc #12
[*] debugger hit 00441640 cc #13
[*] debugger hit 0040ac20 cc #14
[*] debugger hit 0040aa10 cc #15
[*] debugger hit 004172a0 cc #16
[*] debugger hit 004136a0 cc #17
[*] debugger hit 004148f0 cc #18
[*] debugger hit 00414210 cc #19
[*] debugger hit 00438f90 cc #20
[*] debugger hit 00419c30 cc #21
[*] debugger hit 00429300 cc #22
[*] debugger hit 0047fe00 cc #23
[*] debugger hit 00462870 cc #24
[*] debugger hit 0040da50 cc #25
[*] debugger hit 0040dab0 cc #26
[*] debugger hit 0047f550 cc #27
[*] debugger hit 00429110 cc #28
[*] debugger hit 0042ace0 cc #29
[*] debugger hit 004156d0 cc #30
[*] debugger hit 0040d980 cc #31
[*] debugger hit 0040daf0 cc #32
[*] debugger hit 00418090 cc #33
[*] debugger hit 0047fd00 cc #34
[*] debugger hit 0040da20 cc #35
[*] debugger hit 0040d9a0 cc #36
[*] debugger hit 0047da80 cc #37
```

Successfully connected to MySQL server at localhost. | Process Stalker | MySQL: localhost | PyDbg: localhost | uDraw: NONE | User: None

Code Coverage Diffing

- Set "Filter" as a filter tag, stalk "About"



PIDA Modules

# Func	# BB	PIDA Module
3251	28827	notepad++.exe

Process
taskhost.exe
audiodg.exe
cmd.exe
conhost.exe
python.exe
notepad++.exe

rage Depth
nctions
sic Blocks
tore BPs [✓] Heavy [✓] Unhandled Only

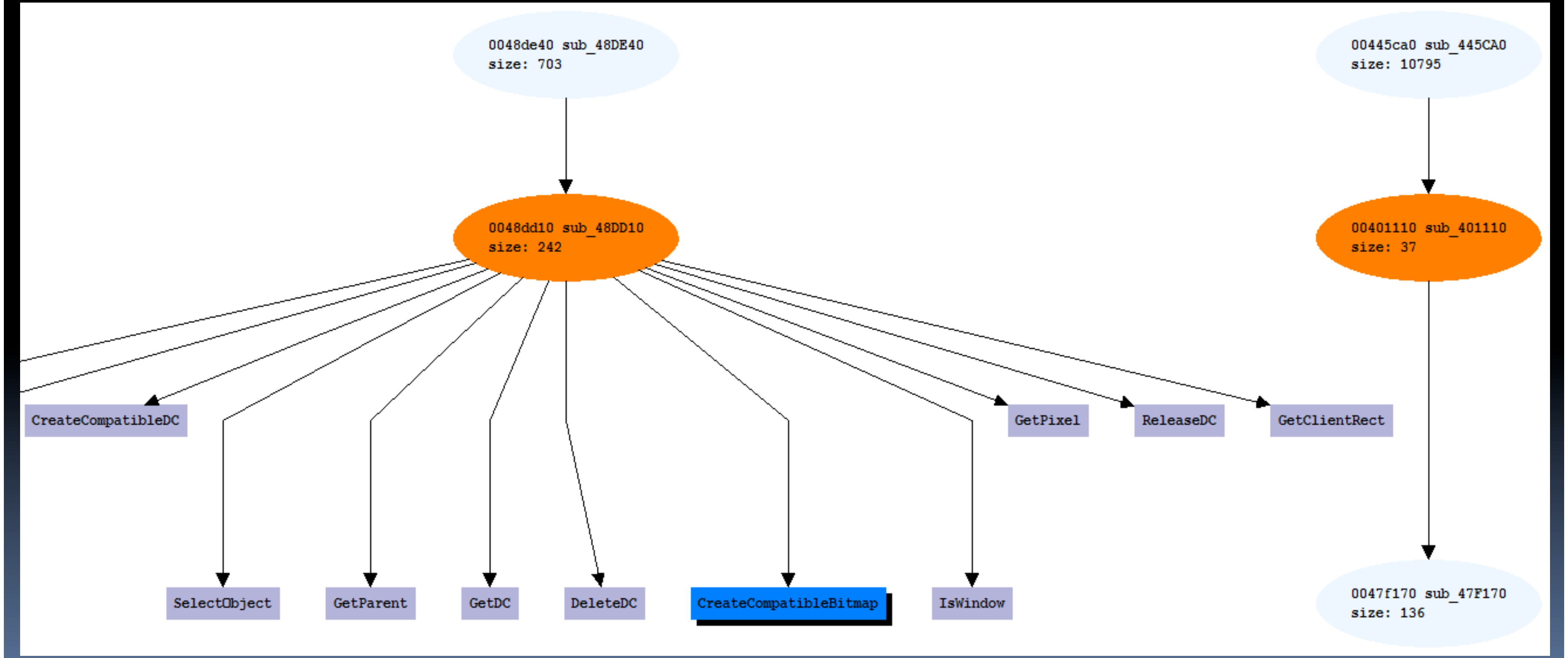
```
[*] Loading 0x744b0000 \Windows\System32\xm
[*] Loading 0x6cc40000 \Windows\System32\UI
[*] Loading 0x77b00000 \Windows\System32\ps
[*] Loading 0x75aa0000 \Windows\System32\sx
[*] Loading 0x6c630000 \Windows\System32\Str
[*] Loading 0x75950000 \Windows\System32\se
[*] Loading 0x75970000 \Windows\System32\ss
[*] Loading 0x6f100000 \Windows\System32\act
[*] Loading 0x72250000 \Program Files\Inter
[*] Loading 0x6df00000 \Windows\System32\thu
[*] Loading 0x704c0000 \Windows\System32\Sea
[*] Loading 0x6ee10000 \Windows\System32\sh
[*] Loading 0x6cf00000 \Windows\System32\ie
[*] Loading 0x6ec00000 \Windows\System32\ne
[*] Loading 0x6ee40000 \Windows\System32\li
[*] Loading 0x73a40000 \Windows\System32\mp
[*] Loading 0x74430000 \Windows\System32\dr
[*] Loading 0x75180000 \Windows\System32\wi
[*] Loading 0x73b80000 \Windows\System32\nt
[*] Loading 0x6cc00000 \Windows\System32\da
[*] Loading 0x6cbf0000 \Windows\System32\da
[*] Loading 0x741e0000 \Windows\System32\wk
[*] Loading 0x741f0000 \Windows\System32\ne
[*] Loading 0x74020000 \Windows\System32\wi
[*] Loading 0x70390000 \Windows\System32\Po
[*] Loading 0x75dd0000 \Windows\System32\wintrust.dll
[*] Loading 0x73b50000 \Windows\System32\EhStorAPI.dll
[*] debugger hit 00401910 cc #1
[*] debugger hit 00401110 cc #2
[*] debugger hit 0048dd10 cc #3
```

-2 chars 0 byt Ln : 1 Col : 1 Sel : 0 (0 bytes) in 0 ranges Dos\Windows ANSI INS

Successfully connected to MySQL server at localhost. | Process Stalker | MySQL: localhost | PyDbg: localhost | uDraw: NONE | User: None

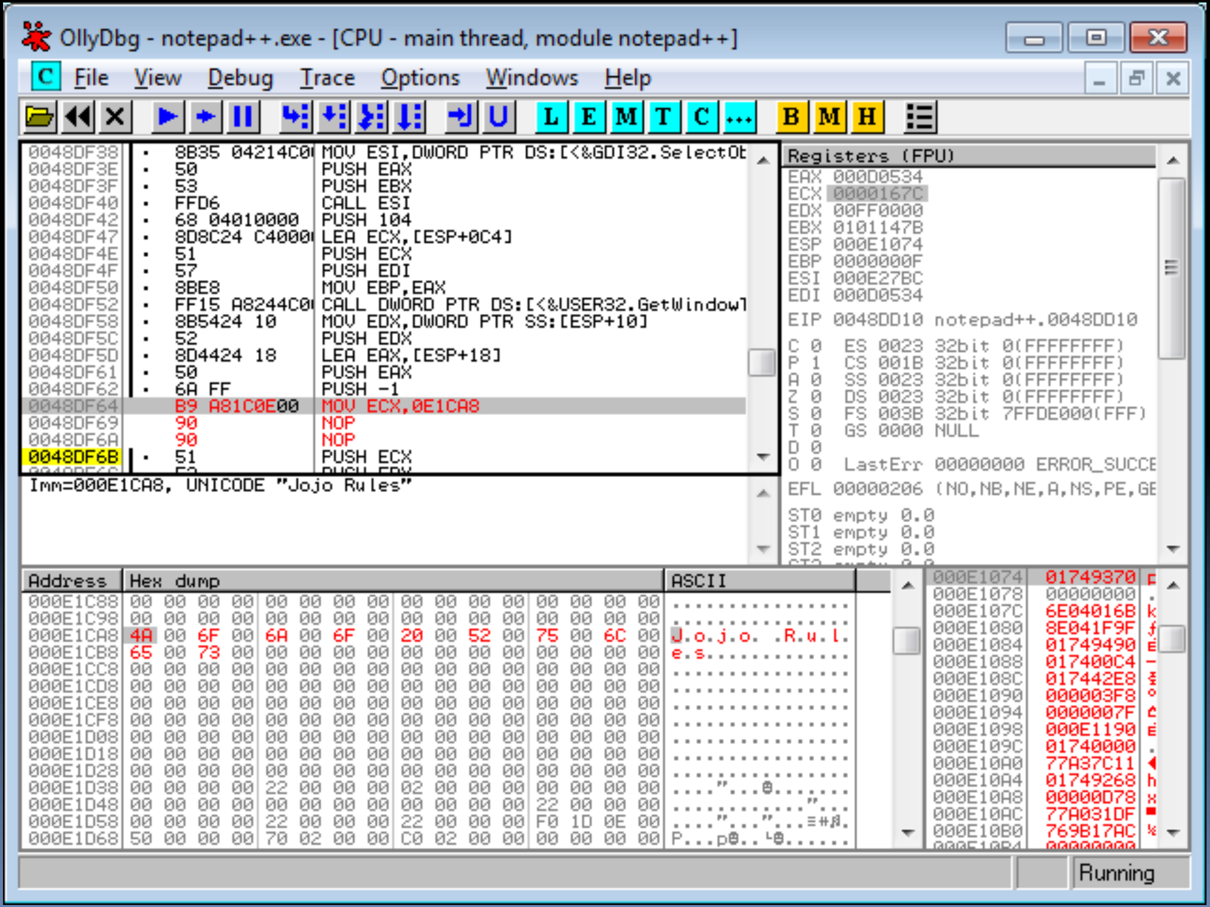
Code Coverage Diffing

- Check the function flow graph in UDraw



Code Coverage Diffing

- Let’s patch the “About” function


OllyDbg - notepad++.exe - [CPU - main thread, module notepad++]
File View Debug Trace Options Windows Help
L E M T C ... B M H

```
0048DF38  .  8B35 04214C0( MOV ESI,DWORD PTR DS:[<&GDI32.SelectOt
0048DF3E  .  50            PUSH EAX
0048DF3F  .  53            PUSH EBX
0048DF40  .  FFD6          CALL ESI
0048DF42  .  68 04010000   PUSH 104
0048DF47  .  8D8C24 C4000( LEA ECX,[ESP+0C4]
0048DF4E  .  51            PUSH ECX
0048DF4F  .  57            PUSH EDI
0048DF50  .  8BE8          MOV EBP,EAX
0048DF52  .  FF15 A8244C0( CALL DWORD PTR DS:[<&USER32.GetWindowl
0048DF58  .  8B5424 10     MOV EDX,DWORD PTR SS:[ESP+10]
0048DF5C  .  52            PUSH EDX
0048DF5D  .  8D4424 18     LEA EAX,[ESP+18]
0048DF61  .  50            PUSH EAX
0048DF62  .  6A FF         PUSH -1
0048DF64     B9 A81C0E00   MOV ECX,0E1CA8
0048DF69     90            NOP
0048DF6A     90            NOP
0048DF6B  .  51            PUSH ECX
Imm=000E1CA8, UNICODE "Jojo Rules"
```

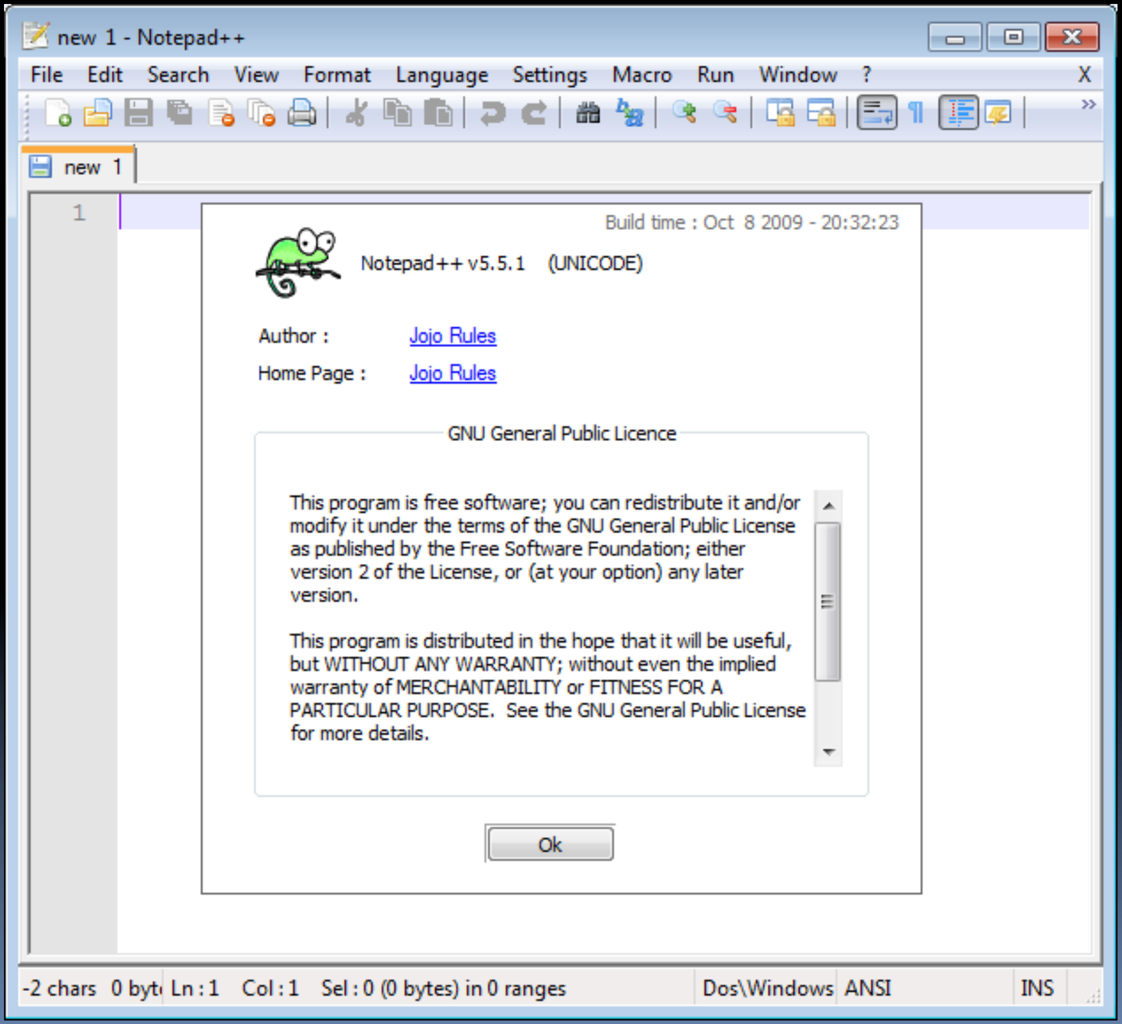
```
Registers (FPU)
EAX 000D0534
ECX 0000167C
EDX 00FF0000
EBX 0101147B
ESP 000E1074
EBP 0000000F
ESI 000E27BC
EDI 000D0534

EIP 0048DD10 notepad++.0048DD10

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDE000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr 00000000 ERROR_SUCCE

EFL 00000206 (NO,NB,NE,A,NS,PE,GE

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
```

```
Address  Hex dump                                         ASCII
000E1C88 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000E1C98 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000E1CA8 4A 00 6F 00 6A 00 6F 00 20 00 52 00 75 00 6C 00 J.o.j.o. .R.u.l.
000E1CB8 65 00 73 00 00 00 00 00 00 00 00 00 00 00 00 00 e.s.............
000E1CC8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000E1CD8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000E1CE8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000E1CF8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000E1D08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000E1D18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000E1D28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000E1D38 00 00 00 00 22 00 00 00 02 00 00 00 00 00 00 00 ....”...☻.......
000E1D48 00 00 00 00 00 00 00 00 00 00 00 00 22 00 00 00 ............”...
000E1D58 00 00 00 00 22 00 00 00 22 00 00 00 F0 1D 0E 00 ....”...”...≡#♬.
000E1D68 50 00 00 00 70 02 00 00 C0 02 00 00 00 00 00 00 P...p☻..└☻......
```

```
000E1074  01749370  p
000E1078  00000000  .
000E107C  6E04016B  k
000E1080  8E041F9F  ƒ
000E1084  01749490  é
000E1088  017400C4  ─
000E108C  017442E8  Φ
000E1090  000003F8  °
000E1094  0000007F  ⌂
000E1098  000E1190  É
000E109C  01740000  .
000E10A0  77A37C11  ◄
000E10A4  01749268  h
000E10A8  00000D78  x
000E10AC  77A031DF  ▀
000E10B0  769B17AC  ¼
```

Running

Code Coverage Diffing


new 1 - Notepad++
File Edit Search View Format Language Settings Macro Run Window ? X
new 1
1
Ok

Build time : Oct 8 2009 - 20:32:23

Notepad++ v5.5.1 (UNICODE)

Author : Jojo Rules

Home Page : Jojo Rules

GNU General Public Licence

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

-2 chars 0 byt Ln : 1 Col : 1 Sel : 0 (0 bytes) in 0 ranges Dos\Windows ANSI INS

Questions/Comments?