



x64dbg Walk Through

Jack Ward

Black Lantern Security

jack.ward@blsgvt.com

Overview

Quick look at using x64dbg to debug, troubleshoot, and reverse engineer simple executables along with a glance at using Ghidra in conjunction with x64dbg

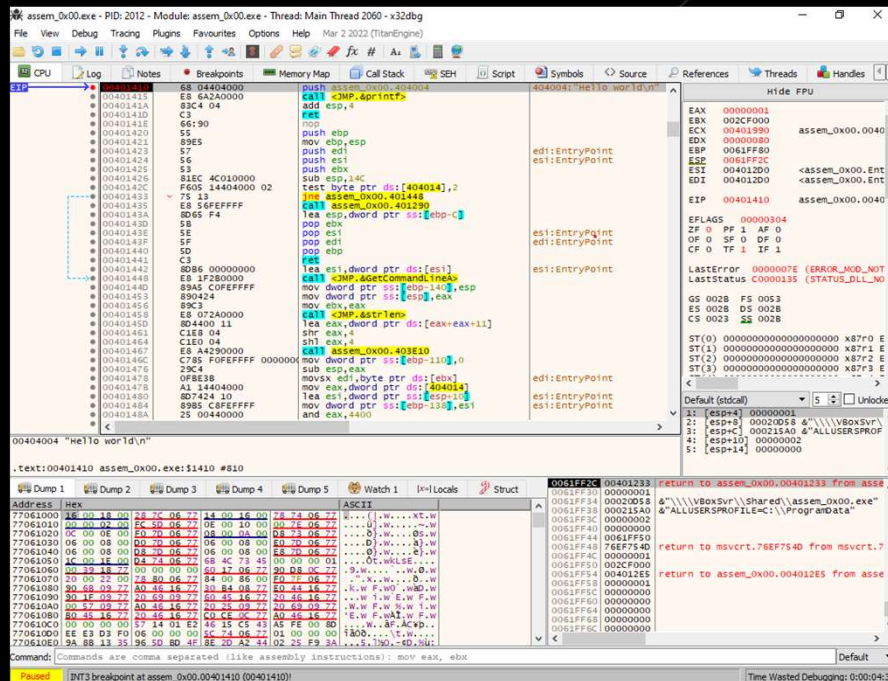
Agenda

- 
- Tools Overview
 - x64dbg
 - Ghidra
 - Workflow
 - Demo
 - Questions



Tools Overview

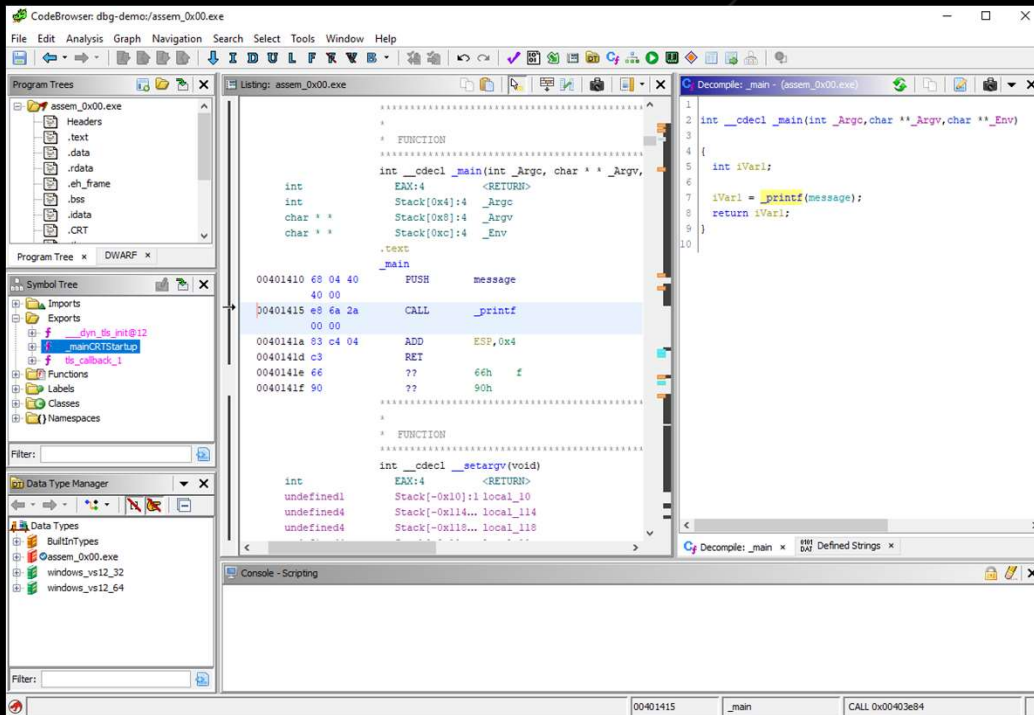
x64dbg



An open-source
x64/x32
debugger
for windows

<https://x64dbg.com/>

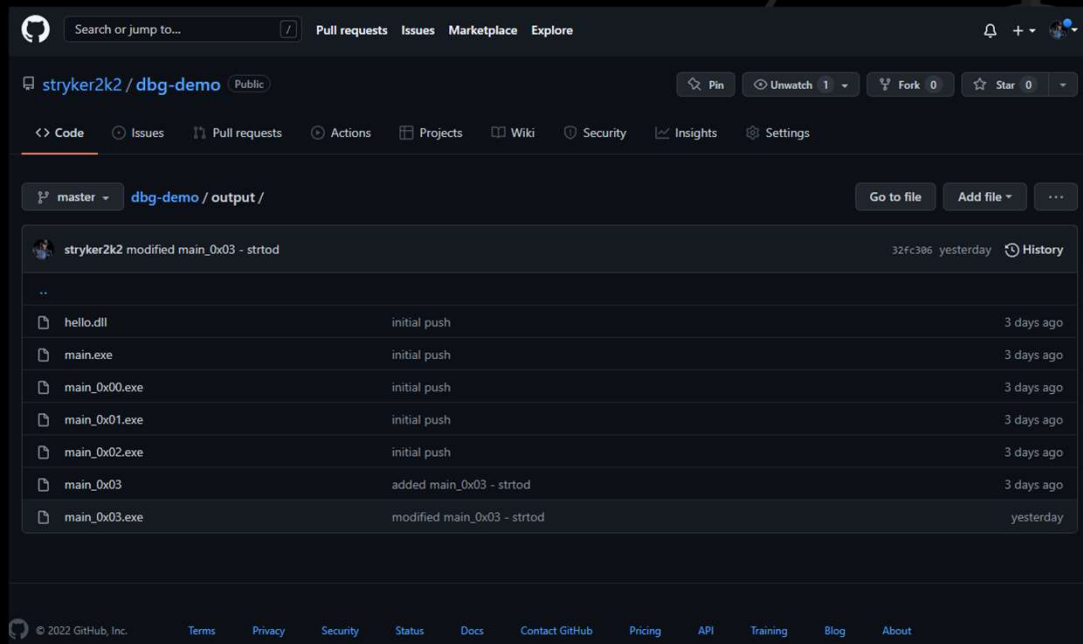
Ghidra



A software reverse engineering (SRE) suite of tools developed by NSA's Research Directorate in support of the Cybersecurity mission

<https://ghidra-sre.org/>

Resources



GitHub repository housing source code, docs, and the executables seen in this demonstration

<https://github.com/stryker2k2/dbg-demo>



x64dbg

x64dbg User Interface

Play Controls
View Tabs
CPU Assembly
Registers
Dumps (Heap)
Stack

The screenshot displays the x64dbg user interface for debugging the process 'assem_0x00.exe'. The main window shows the CPU assembly view, which is currently paused at address 00401410. The assembly code is displayed in a list view, with the current instruction highlighted. The registers window on the right shows the current state of the CPU registers, including EAX, EBX, ECX, EDI, and EIP. The bottom of the interface features a command line and a status bar.

Assembly View (Address 00401410):

```
00401410  E8 6A2A0000  call <JMP.&printf>
00401411  83C4 04      add esp,4
00401412  C3          ret
00401413  66:90       nop
00401414  55          push ebp
00401415  89E5       mov ebp,esp
00401416  57          push edi
00401417  56          push esi
00401418  53          push ebx
00401419  81EC 4C010000 sub esp,14C
0040141A  F605 14404000 02 test byte ptr ds:[404014],2
0040141B  75 13       jne assem_0x00.401448
0040141C  E8 56FEFFFF  call assem_0x00.401290
0040141D  8D65 F4     lea esp,dword ptr ss:[ebp-C]
0040141E  5B          pop ebx
0040141F  5E          pop esi
00401420  5F          pop edi
00401421  5D          pop ebp
00401422  C3          ret
00401423  8DB6 00000000 lea esi,dword ptr ds:[esi]
00401424  E8 1F280000  call <JMP.&GetCommandLineA>
00401425  89A5 C0EFFFFF mov dword ptr ss:[ebp-140],esp
00401426  890424      mov ebx,eax
00401427  89C3       mov ebx,eax
00401428  E8 072A0000  call <JMP.&strlen>
00401429  8D4400 11   lea eax,dword ptr ds:[eax+eax+11]
0040142A  C1E8 04     shr eax,4
0040142B  C1E0 04     shl eax,4
0040142C  E8 A4290000  call assem_0x00.403E10
0040142D  C785 F0EFFFFF 00000000 mov dword ptr ss:[ebp-110],0
0040142E  29C4       sub esp,eax
0040142F  0FBEB3     movsx edi,byte ptr ds:[ebx]
00401430  A1 14404000 mov eax,dword ptr ds:[404014]
00401431  8D7424 10   lea esi,dword ptr ss:[esp+10]
00401432  89B5 C8EFFFFF mov dword ptr ss:[ebp-138],esi
00401433  25 00440000 and eax,4400
```

Registers Window:

Register	Value
EAX	00000001
EBX	002CF000
ECX	00401990
EDX	00000080
EBP	0061FF80
ESP	0061FF2C
ESI	004012D0
EDI	004012D0
EIP	00401410

Stack Window:

Address	Value
00000000	00000000
00000001	00000000
00000002	00000000
00000003	00000000
00000004	00000000
00000005	00000000
00000006	00000000
00000007	00000000
00000008	00000000
00000009	00000000
0000000A	00000000
0000000B	00000000
0000000C	00000000
0000000D	00000000
0000000E	00000000
0000000F	00000000
00000010	00000000
00000011	00000000
00000012	00000000
00000013	00000000
00000014	00000000
00000015	00000000
00000016	00000000
00000017	00000000
00000018	00000000
00000019	00000000
0000001A	00000000
0000001B	00000000
0000001C	00000000
0000001D	00000000
0000001E	00000000
0000001F	00000000
00000020	00000000
00000021	00000000
00000022	00000000
00000023	00000000
00000024	00000000
00000025	00000000
00000026	00000000
00000027	00000000
00000028	00000000
00000029	00000000
0000002A	00000000
0000002B	00000000
0000002C	00000000
0000002D	00000000
0000002E	00000000
0000002F	00000000
00000030	00000000
00000031	00000000
00000032	00000000
00000033	00000000
00000034	00000000
00000035	00000000
00000036	00000000
00000037	00000000
00000038	00000000
00000039	00000000
0000003A	00000000
0000003B	00000000
0000003C	00000000
0000003D	00000000
0000003E	00000000
0000003F	00000000
00000040	00000000
00000041	00000000
00000042	00000000
00000043	00000000
00000044	00000000
00000045	00000000
00000046	00000000
00000047	00000000
00000048	00000000
00000049	00000000
0000004A	00000000
0000004B	00000000
0000004C	00000000
0000004D	00000000
0000004E	00000000
0000004F	00000000
00000050	00000000
00000051	00000000
00000052	00000000
00000053	00000000
00000054	00000000
00000055	00000000
00000056	00000000
00000057	00000000
00000058	00000000
00000059	00000000
0000005A	00000000
0000005B	00000000
0000005C	00000000
0000005D	00000000
0000005E	00000000
0000005F	00000000
00000060	00000000
00000061	00000000
00000062	00000000
00000063	00000000
00000064	00000000
00000065	00000000
00000066	00000000
00000067	00000000
00000068	00000000
00000069	00000000
0000006A	00000000
0000006B	00000000
0000006C	00000000
0000006D	00000000
0000006E	00000000
0000006F	00000000
00000070	00000000
00000071	00000000
00000072	00000000
00000073	00000000
00000074	00000000
00000075	00000000
00000076	00000000
00000077	00000000
00000078	00000000
00000079	00000000
0000007A	00000000
0000007B	00000000
0000007C	00000000
0000007D	00000000
0000007E	00000000
0000007F	00000000
00000080	00000000
00000081	00000000
00000082	00000000
00000083	00000000
00000084	00000000
00000085	00000000
00000086	00000000
00000087	00000000
00000088	00000000
00000089	00000000
0000008A	00000000
0000008B	00000000
0000008C	00000000
0000008D	00000000
0000008E	00000000
0000008F	00000000
00000090	00000000
00000091	00000000
00000092	00000000
00000093	00000000
00000094	00000000
00000095	00000000
00000096	00000000
00000097	00000000
00000098	00000000
00000099	00000000
0000009A	00000000
0000009B	00000000
0000009C	00000000
0000009D	00000000
0000009E	00000000
0000009F	00000000
000000A0	00000000
000000A1	00000000
000000A2	00000000
000000A3	00000000
000000A4	00000000
000000A5	00000000
000000A6	00000000
000000A7	00000000
000000A8	00000000
000000A9	00000000
000000AA	00000000
000000AB	00000000
000000AC	00000000
000000AD	00000000
000000AE	00000000
000000AF	00000000
000000B0	00000000
000000B1	00000000
000000B2	00000000
000000B3	00000000
000000B4	00000000
000000B5	00000000
000000B6	00000000
000000B7	00000000
000000B8	00000000
000000B9	00000000
000000BA	00000000
000000BB	00000000
000000BC	00000000
000000BD	00000000
000000BE	00000000
000000BF	00000000
000000C0	00000000
000000C1	00000000
000000C2	00000000
000000C3	00000000
000000C4	00000000
000000C5	00000000
000000C6	00000000
000000C7	00000000
000000C8	00000000
000000C9	00000000
000000CA	00000000
000000CB	00000000
000000CC	00000000
000000CD	00000000
000000CE	00000000
000000CF	00000000
000000D0	00000000
000000D1	00000000
000000D2	00000000
000000D3	00000000
000000D4	00000000
000000D5	00000000
000000D6	00000000
000000D7	00000000
000000D8	00000000
000000D9	00000000
000000DA	00000000
000000DB	00000000
000000DC	00000000
000000DD	00000000
000000DE	00000000
000000DF	00000000
000000E0	00000000
000000E1	00000000
000000E2	00000000
000000E3	00000000
000000E4	00000000
000000E5	00000000
000000E6	00000000
000000E7	00000000
000000E8	00000000
000000E9	00000000
000000EA	00000000
000000EB	00000000
000000EC	00000000
000000ED	00000000
000000EE	00000000
000000EF	00000000
000000F0	00000000
000000F1	00000000
000000F2	00000000
000000F3	00000000
000000F4	00000000
000000F5	00000000
000000F6	00000000
000000F7	00000000
000000F8	00000000
000000F9	00000000
000000FA	00000000
000000FB	00000000
000000FC	00000000
000000FD	00000000
000000FE	00000000
000000FF	00000000

Stack Window:

Address	Value
0061FF2C	00401233
0061FF30	00000001
0061FF34	00020D58
0061FF38	000215A0
0061FF3C	00000002
0061FF40	00000000
0061FF44	0061FF50
0061FF48	76EF754D
0061FF4C	00000001
0061FF50	002CF000
0061FF54	004012E5
0061FF58	00000001
0061FF5C	00000000
0061FF60	00000000
0061FF64	00000000
0061FF68	00000000
0061FF6C	00000000

Command Line: Commands are comma separated (like assembly instructions): mov eax, ebx

Status Bar: Paused | INT3 breakpoint at assem_0x00.00401410 (00401410) | Time Wasted Debugging: 0:00:04:32

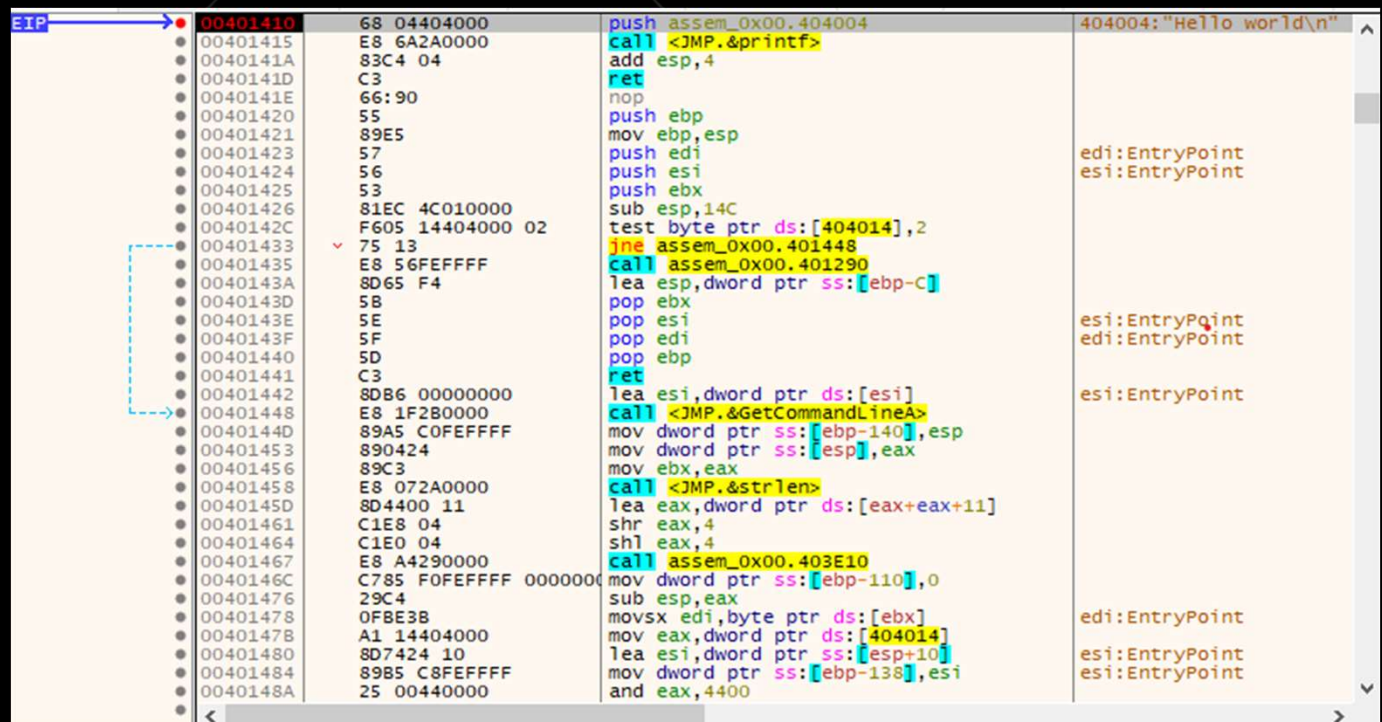
x64dbg User Interface

Play Controls
View Tabs



x64dbg User Interface

CPU Assembly



Address	Disassembly	Comment
00401410	68 04404000 push assem_0x00.404004	404004:"Hello world\n"
00401415	E8 6A2A0000 call <JMP.&printf>	
0040141A	83C4 04 add esp,4	
0040141D	C3 ret	
0040141E	66:90 nop	
00401420	55 push ebp	
00401421	89E5 mov ebp,esp	
00401423	57 push edi	edi:EntryPoint
00401424	56 push esi	esi:EntryPoint
00401425	53 push ebx	
00401426	81EC 4C010000 sub esp,14C	
0040142C	F605 14404000 02 test byte ptr ds:[404014],2	
00401433	75 13 jne assem_0x00.401448	
00401435	E8 56FEFFFF call assem_0x00.401290	
0040143A	8D65 F4 lea esp,dword ptr ss:[ebp-C]	
0040143D	5B pop ebx	
0040143E	5E pop esi	esi:EntryPoint
0040143F	5F pop edi	edi:EntryPoint
00401440	5D pop ebp	
00401441	C3 ret	
00401442	8DB6 00000000 lea esi,dword ptr ds:[esi]	esi:EntryPoint
00401448	E8 1F2B0000 call <JMP.&GetCommandLineA>	
0040144D	89A5 C0FEFFFF mov dword ptr ss:[ebp-140],esp	
00401453	890424 mov dword ptr ss:[esp],eax	
00401456	89C3 mov ebx,eax	
00401458	E8 072A0000 call <JMP.&strlen>	
0040145D	8D4400 11 lea eax,dword ptr ds:[eax+eax+11]	
00401461	C1E8 04 shr eax,4	
00401464	C1E0 04 shl eax,4	
00401467	E8 A4290000 call assem_0x00.403E10	
0040146C	C785 F0FEFFFF 000000 mov dword ptr ss:[ebp-110],0	
00401476	29C4 sub esp,eax	
00401478	0FB3B8 movsx edi,byte ptr ds:[ebx]	edi:EntryPoint
0040147B	A1 14404000 mov eax,dword ptr ds:[404014]	
00401480	8D7424 10 lea esi,dword ptr ss:[esp+10]	esi:EntryPoint
00401484	89B5 C8FEFFFF mov dword ptr ss:[ebp-138],esi	esi:EntryPoint
0040148A	25 00440000 and eax,4400	

x64dbg User Interface

Registers

```
Hide FPU

EAX 00000001
EBX 002CF000
ECX 00401990      assem_0x00.0040
EDX 00000080
EBP 0061FF80
ESP 0061FF2C
ESI 004012D0      <assem_0x00.Ent
EDI 004012D0      <assem_0x00.Ent

EIP 00401410      assem_0x00.0040

EFLAGS 00000304
ZF 0 PF 1 AF 0
OF 0 SF 0 DF 0
CF 0 TF 1 IF 1

LastError 0000007E (ERROR_MOD_NOT
LastStatus C0000135 (STATUS_DLL_NO

GS 002B FS 0053
ES 002B DS 002B
CS 0023 SS 002B

ST(0) 00000000000000000000 x87r0 E
ST(1) 00000000000000000000 x87r1 E
ST(2) 00000000000000000000 x87r2 E
ST(3) 00000000000000000000 x87r3 E

Default (stdcall) 5 Unlocked
1: [esp+4] 00000001
2: [esp+8] 00020D58 &"\\\\VBoxSvr\\
3: [esp+C] 000215A0 &"ALLUSERSPROF
4: [esp+10] 00000002
5: [esp+14] 00000000
```

EAX: Math and Return Values

EBX:
Base index (for use with arrays)

ECX: Counter

EDX: Math

EBP: Base Pointer

ESP: Stack Pointer

ESI/EDI: Memory Transfer

Dumps (Heap)

Dump 1	Dump 2	Dump 3	Dump 4	Dump 5	Watch 1	[x=] Locals	Struct
Address	Hex					ASCII	
77061000	16 00 18 00	28 7C 06 77	14 00 16 00	78 74 06 77	*...{ .w...xt.w		
77061010	00 00 02 00	FC 5D 06 77	0E 00 10 00	00 7E 06 77	...ü}.w...~.w		
77061020	0C 00 0E 00	F0 7D 06 77	08 00 0A 00	D8 73 06 77	...ð}.w...ø.s.w		
77061030	06 00 08 00	D0 7D 06 77	06 00 08 00	E0 7D 06 77	...ð}.w...ä}.w		
77061040	06 00 08 00	D8 7D 06 77	06 00 08 00	E8 7D 06 77	...ø}.w...è}.w		
77061050	1C 00 1E 00	D4 74 06 77	6B 4C 73 45	00 00 00 01	...øt.wkLse...		
77061060	00 39 18 77	00 00 00 00	60 17 06 77	90 D8 0C 77	.9.w...w.ø.w		
77061070	20 00 22 00	78 80 06 77	84 00 86 00	F0 7F 06 77	".x..w...ð..w		
77061080	90 68 09 77	A0 46 16 77	30 84 08 77	E0 44 16 77	.k.w F.w0'.wäd.w		
77061090	90 1F 09 77	20 69 09 77	60 45 16 77	20 46 16 77	...w i.w'E.w F.w		
770610A0	00 57 09 77	A0 46 16 77	20 25 09 77	20 69 09 77	.W.w F.w %.w i.w		
770610B0	80 45 16 77	20 46 16 77	C0 CE 0C 77	A0 46 16 77	*E.w F.wAi.w F.w		
770610C0	00 00 00 00	57 14 01 E2	46 15 C5 43	A5 FE 00 8D	...w...âF.Ac*þ..		
770610D0	E3 D3 F0	06 00 00 00	5C 74 06 77	01 00 00 00	iäö...t.w...		
770610E0	9A 88 13 35	96 5D 8D 4F	8E 2D A2 44	02 25 F9 3A	...s.%ø...ç.%ü:		

x64dbg User Interface

Stack

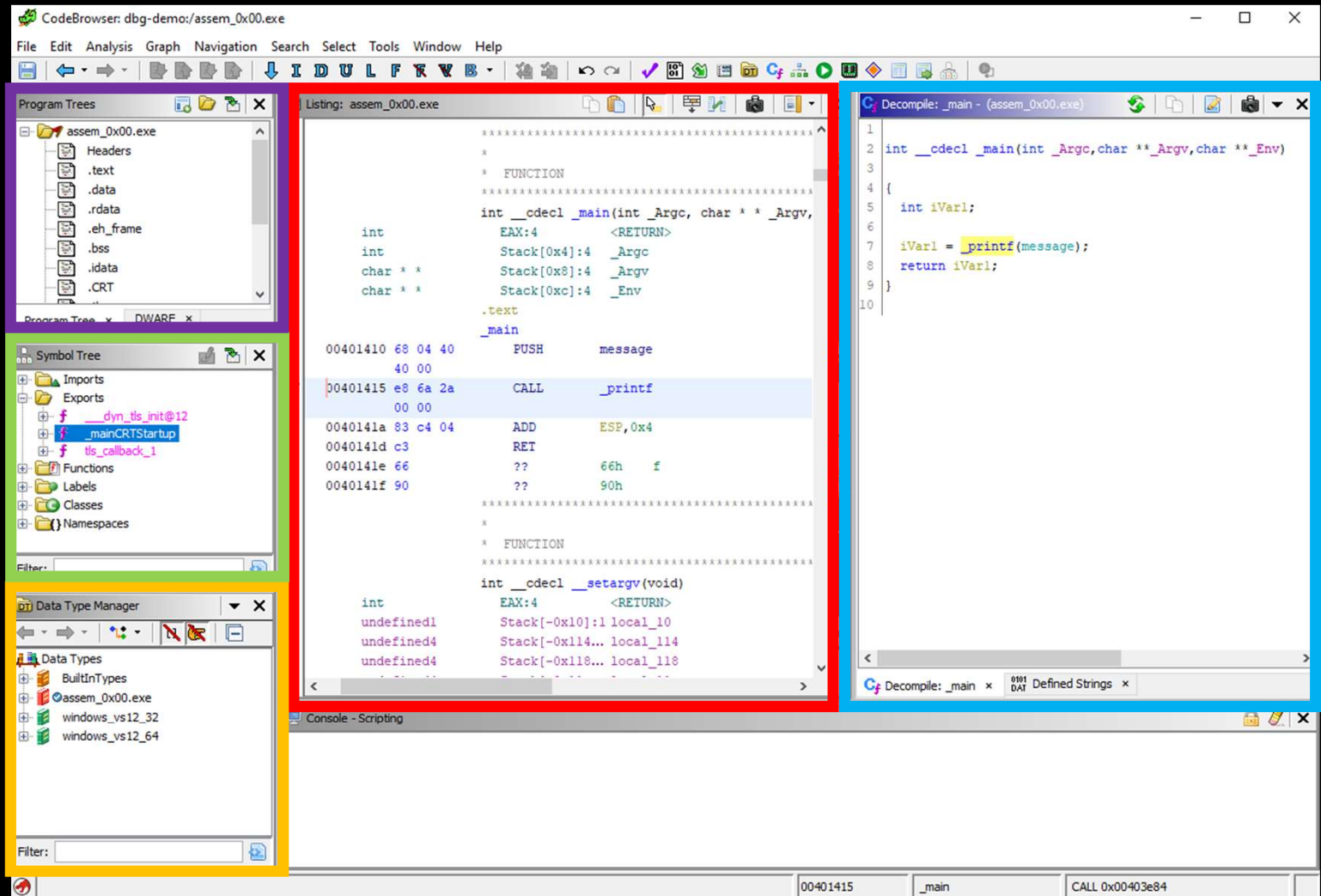
0061FF2C	00401233	return to assem_0x00.00401233 from asse
0061FF30	00000001	
0061FF34	00020D58	&"\\\\\\VBoxSvr\\Shared\\assem_0x00.exe"
0061FF38	000215A0	&"ALLUSERSPROFILE=C:\\ProgramData"
0061FF3C	00000002	
0061FF40	00000000	
0061FF44	0061FF50	
0061FF48	76EF754D	return to msvcrt.76EF754D from msvcrt.7
0061FF4C	00000001	
0061FF50	002CF000	
0061FF54	004012E5	return to assem_0x00.004012E5 from asse
0061FF58	00000001	
0061FF5C	00000000	
0061FF60	00000000	
0061FF64	00000000	
0061FF68	00000000	
0061FF6C	00000000	



Ghidra

Ghidra User Interface

Listing (ASM)
Decompiler (C)
Program Trees
Symbol Tree
Data Types



Assembly VS Decompiler

Listing (ASM):

- push message
- call _printf

Decompiler (C)

- call _printf
- pass message

The screenshot shows a debugger window with two panes. The left pane, titled 'Listing: assem_0x00.exe', displays assembly code for the function `__cdecl _main`. The right pane, titled 'Decompile: _main - (assem_0x00.exe)', shows the corresponding C code. In the assembly pane, the instruction `PUSH message` at address `00401410` is highlighted with a red box, and the instruction `CALL _printf` at address `00401415` is highlighted with a blue box. In the decompiled code pane, the line `iVar1 = printf(message);` is highlighted with a red box, showing the C equivalent of the assembly instructions.

```
Listing: assem_0x00.exe
*****
* FUNCTION
*****
int __cdecl _main(int _Argc, char * *_Argv,
char * *_Env)
char * *_Env
char * *_Argv
Stack[0x8]:4
Stack[0x4]:4
EAX:4
<RETURN>
int
int
.text
_main
00401410 68 04 40 PUSH message
40 00
00401415 e8 6a 2a CALL _printf
00 00
0040141a 83 c4 04 ADD ESP,0x4
0040141d c3 RET
0040141e 66 ?? 66h f
0040141f 90 ?? 90h
*****
* FUNCTION
*****
int __cdecl __setargv(void)
int
EAX:4
<RETURN>
undefined1 Stack[-0x10]:1 local_10
undefined4 Stack[-0x114... local_114
undefined4 Stack[-0x118... local_118
*****

Decompile: _main - (assem_0x00.exe)
1
2 int __cdecl _main(int _Argc, char **_Argv, char **_Env)
3
4 {
5     int iVar1;
6
7     iVar1 = printf(message);
8     return iVar1;
9 }
10
```



x64dbg
Ghidra
Workflow

```
x64dbg
Dynamic
00401410
"Hello World"
```

Ghidra Static

00401410
(message)

EIP	Address	Disassembly	Comment
	00401410	push assem_0x00.404004	404004: "Hello world\n"
	00401415	call <JMP.&printf>	
	0040141A	add esp,4	
	0040141D	ret	
	0040141E	66:90	
	00401420	55	
	00401421	89E5	
	00401423	57	
	00401424	56	
	00401425	53	
	00401426	81EC 4C010000	
	0040142C	F605 14404000 02	
	00401433	75 13	
	00401435	E8 56FEFFFF	
	0040143A	8D65 F4	
	0040143D	5B	
	0040143E	5E	
	0040143F	5F	
	00401440	5D	
	00401441	C3	
	00401442	8DB6 00000000	
	00401448	E8 1F280000	
	0040144D	8B4E C0F5FFFF	
		push esi, dword ptr ds:[esi]	esi:EntryPoint
		call <JMP.&GetCommandLineA>	esi:EntryPoint
		mov dword ptr ss:[ebp+104], esp	esi:EntryPoint

The image displays two side-by-side windows from a debugger. The left window, titled "Listing: assem_0x00.exe", shows the assembly code for a function named `__cdecl __main`. It includes variable declarations for `_Argc`, `_Argv`, and `_Env`, and a call to `_printf`. The right window, titled "Decompile: _main - (assem_0x00.exe)", shows the corresponding C code. In both windows, the line where `iVar1` is assigned the result of `_printf(message)` is highlighted with a red rectangle.



DEMO



Questions?

Jack Ward

Black Lantern Security

jack.ward@blsgvt.com