x64dbg
Walk Through

Jack Ward
Black Lantern Security
jack.ward@blsgvt.com

Overview

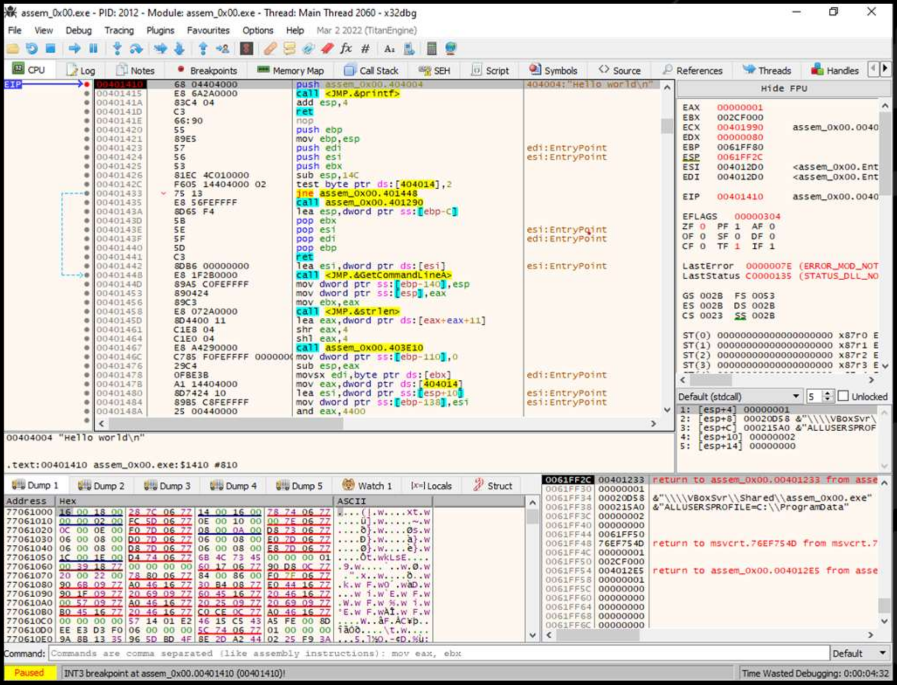
Quick look at using x64dbg to debug, troubleshoot, and reverse engineer simple executables along with a glance at using Ghidra in conjunction with x64dbg

Agenda

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

Tools Overview

x64dbg



assem_0x00.exe - PID: 2012 - Module: assem_0x00.exe - Thread: Main Thread 2060 - x32dbg

Mar 2 2022 (TitanEngine)

```
EIP  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  0FBE3B               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

00404004 "Hello world\n"

.text:00401410 assem_0x00.exe:$1410 #810
```

```
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) 0000000000000000000 x87r0 E
ST(1) 0000000000000000000 x87r1 E
ST(2) 0000000000000000000 x87r2 E
ST(3) 0000000000000000000 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
```

```
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....Øs.w
77061030  06 00 08 00 D0 7D 06 77 06 00 08 00 E0 7D 06 77  ....Ð}.w....à}.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  ....Ôt.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 6B 09 77 A0 46 16 77 30 B4 08 77 E0 44 16 77  .k.w F.w0´.wàD.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  B0 45 16 77 20 46 16 77 C0 CE 0C 77 A0 46 16 77  °E.w F.wÀÎ.w F.w
770610C0  00 00 00 00 57 14 01 E2 46 15 C5 43 A5 FE 00 8D  ....W..âF.ÅC¥þ..
770610D0  EE E3 D3 F0 06 00 00 00 5C 74 06 77 01 00 00 00  îãÓð....\t.w....
770610E0  9A 8B 13 35 96 5D BD 4F 8E 2D A2 44 02 25 F9 3A  ...5.]½O.-¢D.%ù:
```

```
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
```

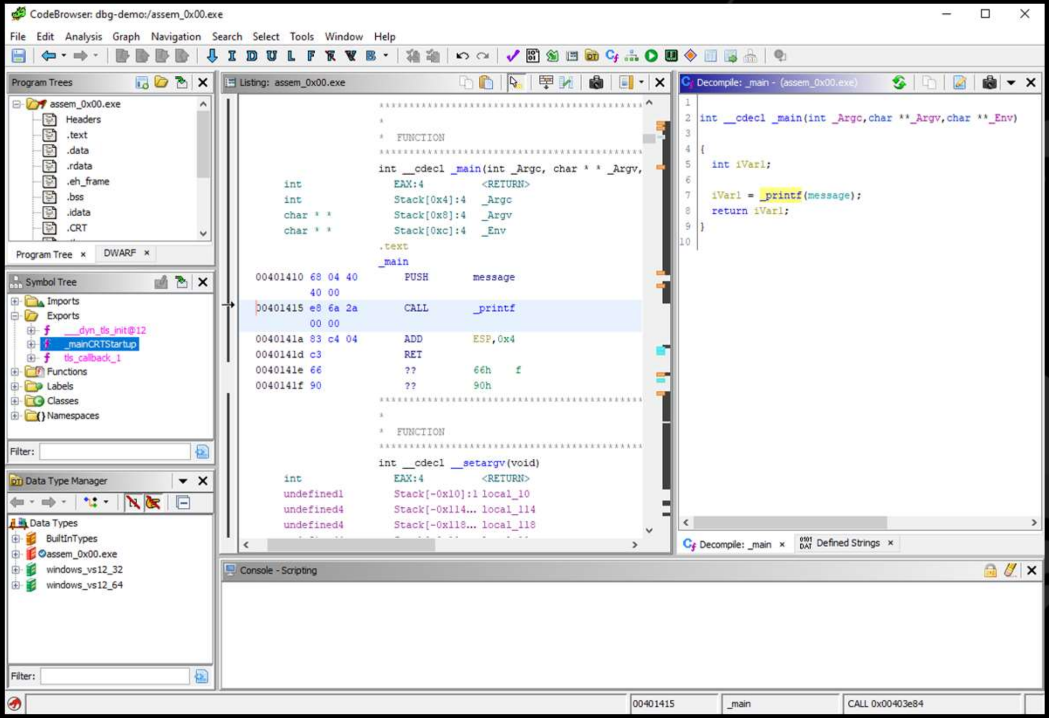
Command: Commands are comma separated (like assembly instructions): mov eax, ebx

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

An open-source x64/x32 debugger for windows

https://x64dbg.com/

Ghidra



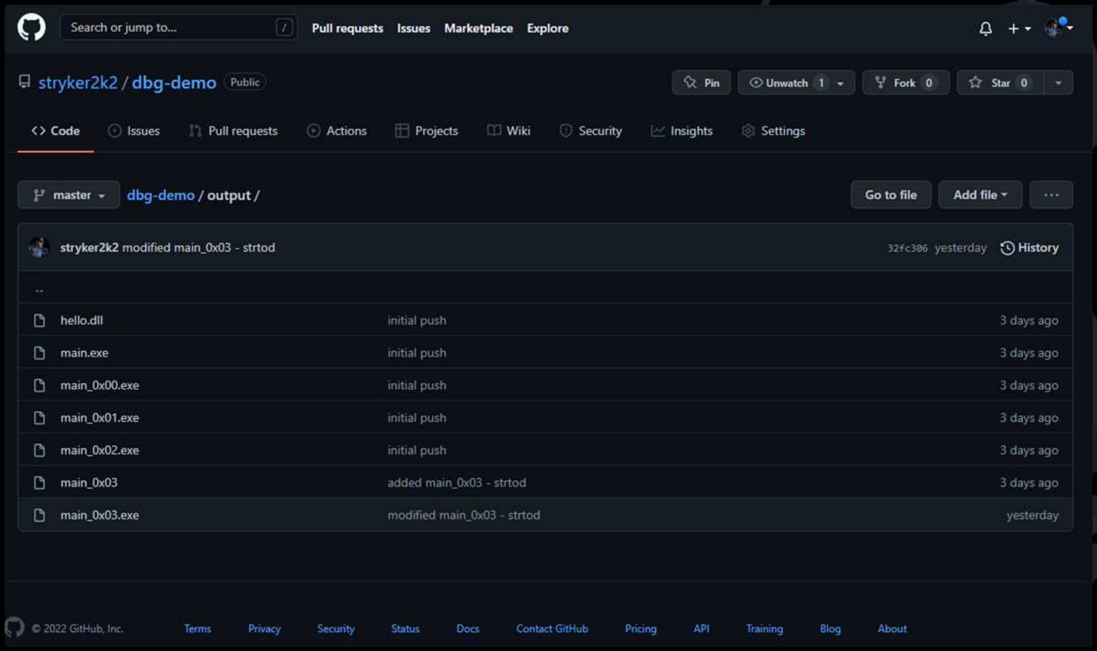
```
                    ******************************************
                    *
                    *  FUNCTION
                    ******************************************
                    int __cdecl _main(int _Argc, char * * _Argv,
     int               EAX:4          <RETURN>
     int               Stack[0x4]:4   _Argc
     char * *          Stack[0x8]:4   _Argv
     char * *          Stack[0xc]:4   _Env
                    .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
```

```
int __cdecl _main(int _Argc,char **_Argv,char **_Env)

{
  int iVar1;

  iVar1 = _printf(message);
  return iVar1;
}
```

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



stryker2k2 modified main_0x03 - strtod 32fc306 yesterday

..		
hello.dll	initial push	3 days ago
main.exe	initial push	3 days ago
main_0x00.exe	initial push	3 days ago
main_0x01.exe	initial push	3 days ago
main_0x02.exe	initial push	3 days ago
main_0x03	added main_0x03 - strtod	3 days ago
main_0x03.exe	modified main_0x03 - strtod	yesterday

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

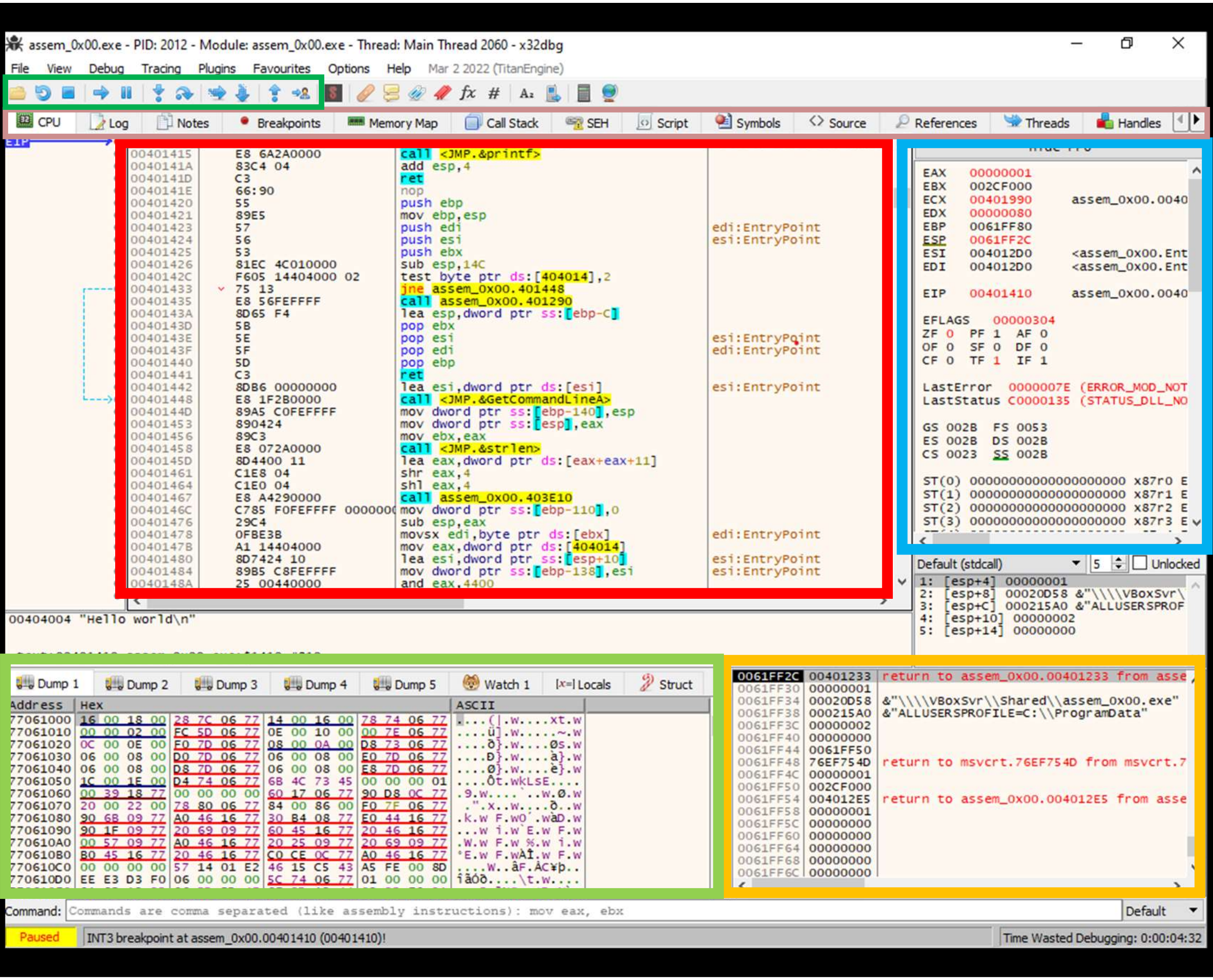
View Tabs

CPU Assembly

Registers

Dumps (Heap)

Stack



```
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 0000000    mov dword ptr ss:[ebp-110],0
00401476    29C4                     sub esp,eax
00401478    0FBE3B                   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
```

```
00404004 "Hello world\n"
```

```
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
```

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

```
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....Øs.w
77061030  06 00 08 00 D0 7D 06 77 06 00 08 00 E0 7D 06 77  ....Ð}.w....à}.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  ....Ôt.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 6B 09 77 A0 46 16 77 30 B4 08 77 E0 44 16 77  .k.w F.w0´.wàD.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  B0 45 16 77 20 46 16 77 C0 CE 0C 77 A0 46 16 77  °E.w F.wÀÎ.w F.w
770610C0  00 00 00 00 57 14 01 E2 46 15 C5 43 A5 FE 00 8D  ....W..âF.ÅC¥þ..
770610D0  EE E3 D3 F0 06 00 00 00 5C 74 06 77 01 00 00 00  îãÓð....\t.w....
```

```
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
```

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

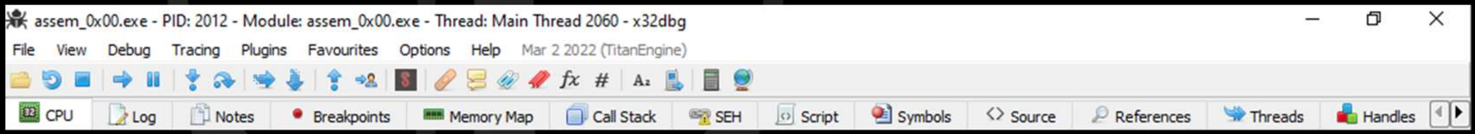
INT3 breakpoint at assem_0x00.00401410 (00401410)!

Time Wasted Debugging: 0:00:04:32

x64dbg User Interface

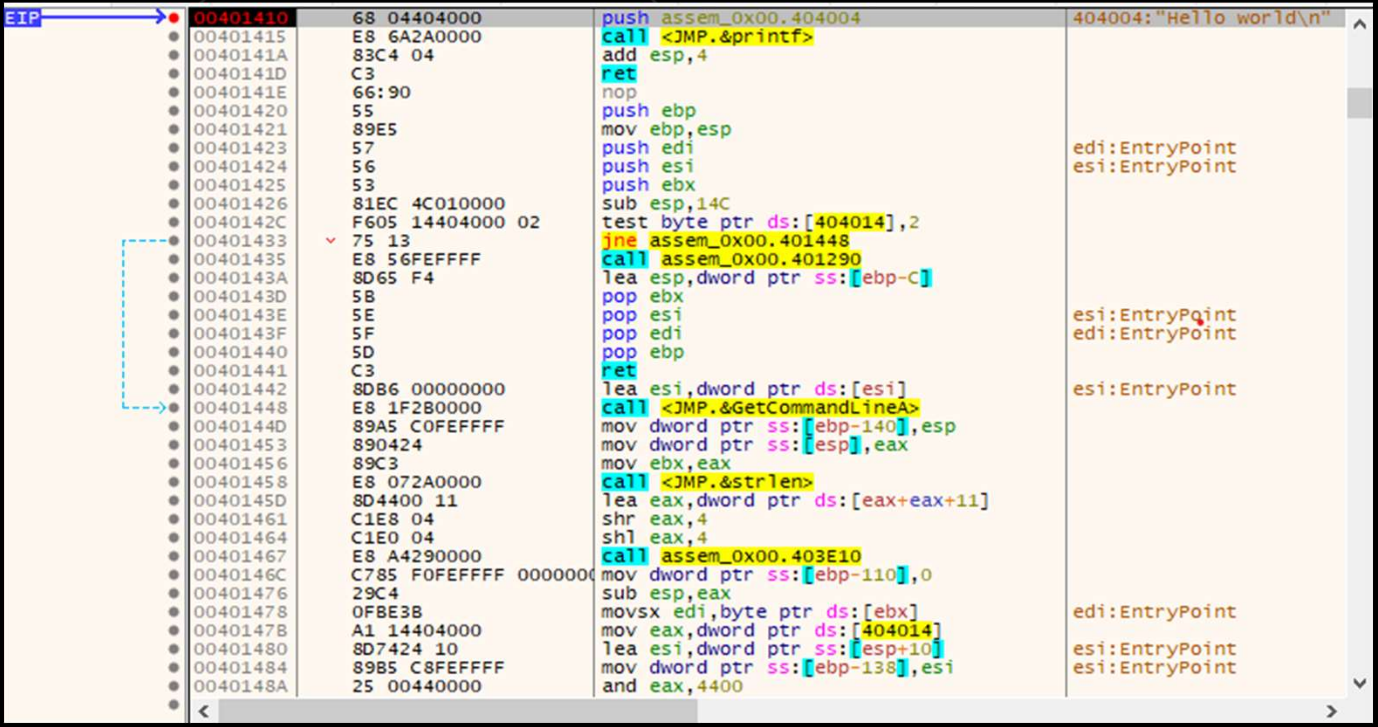
Play Controls

View Tabs



x64dbg User Interface

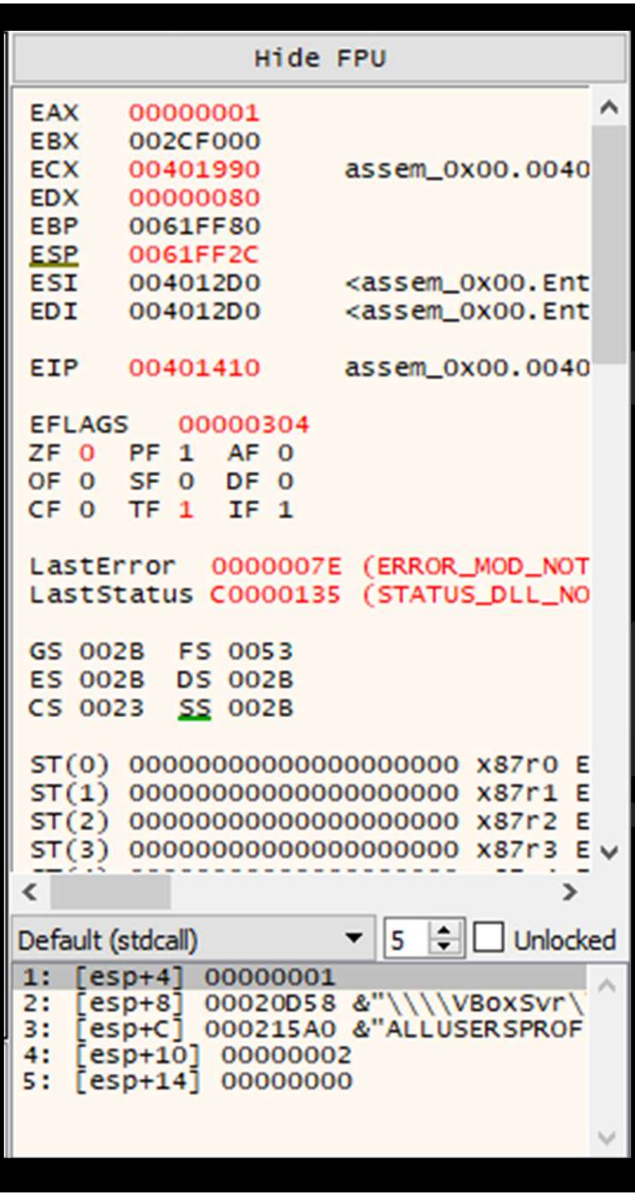
CPU Assembly



```
EIP ──> 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  0FBE3B                   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



```
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

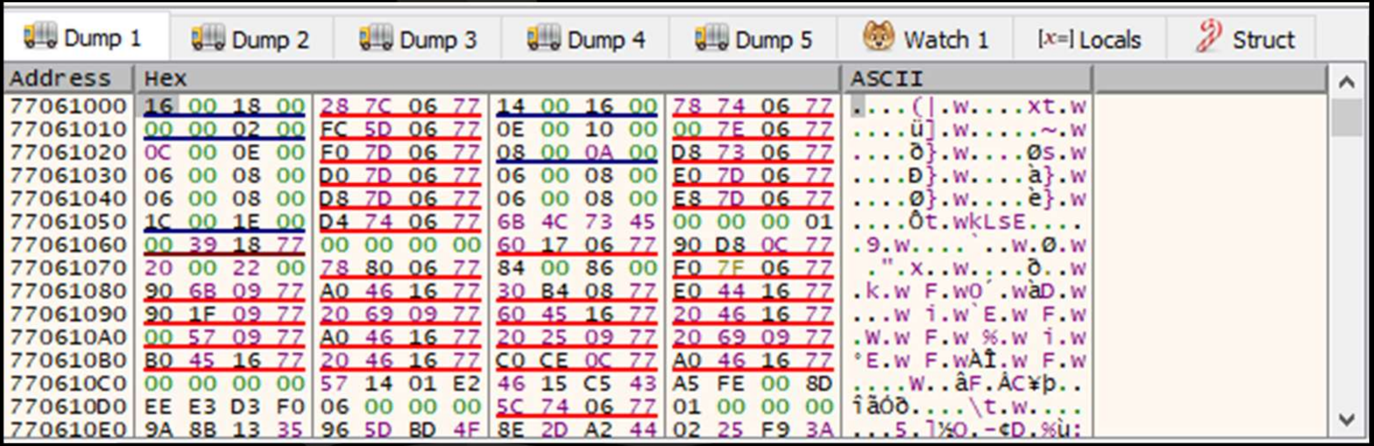
EBP: Base Pointer

ESP: Stack Pointer

ESI/EDI: Memory Transfer

x64dbg User Interface

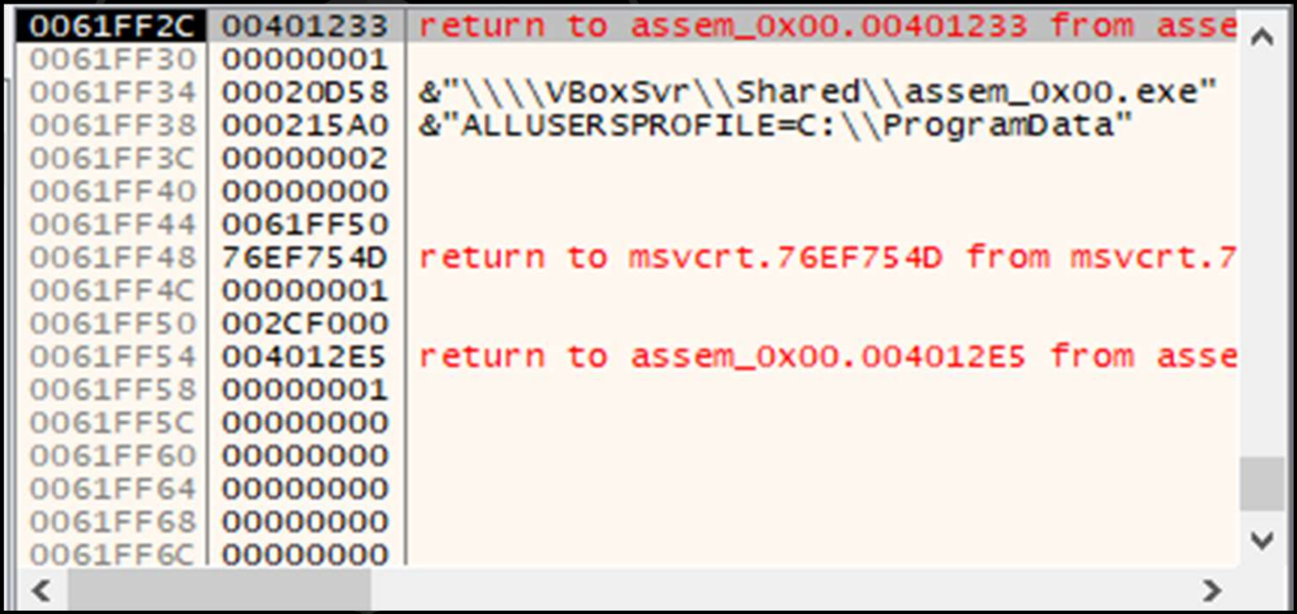
Dumps (Heap)



```
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....Øs.w
77061030  06 00 08 00  D0 7D 06 77  06 00 08 00  E0 7D 06 77  ....Ð}.w....à}.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  ....Ôt.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 6B 09 77  A0 46 16 77  30 B4 08 77  E0 44 16 77  .k.w F.w0´.wàD.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  B0 45 16 77  20 46 16 77  C0 CE 0C 77  A0 46 16 77  °E.w F.wÀÎ.w F.w
770610C0  00 00 00 00  57 14 01 E2  46 15 C5 43  A5 FE 00 8D  ....W..âF.ÅC¥þ..
770610D0  EE E3 D3 F0  06 00 00 00  5C 74 06 77  01 00 00 00  îãÓð....\t.w....
770610E0  9A 8B 13 35  96 5D BD 4F  8E 2D A2 44  02 25 F9 3A  ...5.]½O.-¢D.%ù:
```

x64dbg User Interface

Stack

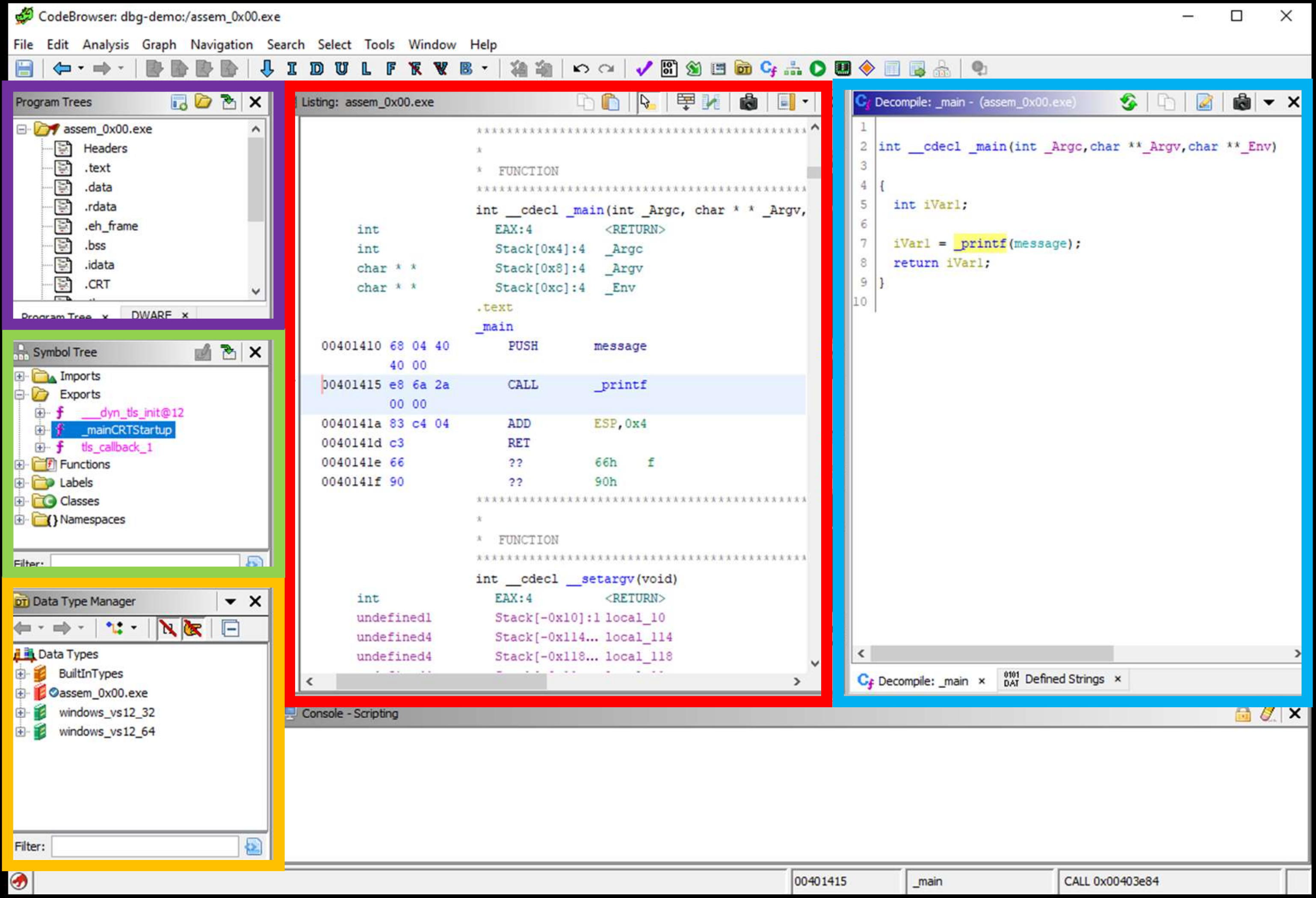


```
0061FF2C  00401233  return to assem_0x00.00401233 from asse
0061FF30  00000001
0061FF34  00020D58  &"\\\\VBoxSvr\\Shared\\assem_0x00.exe"
0061FF38  000215A0  &"ALLUSERSPROFILE=C:\\ProgramData"
0061FF3C  00000002
0061FF40  00000000
0061FF44  0061FF50
0061FF48  76EF754D  return to msvcrt.76EF754D from 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



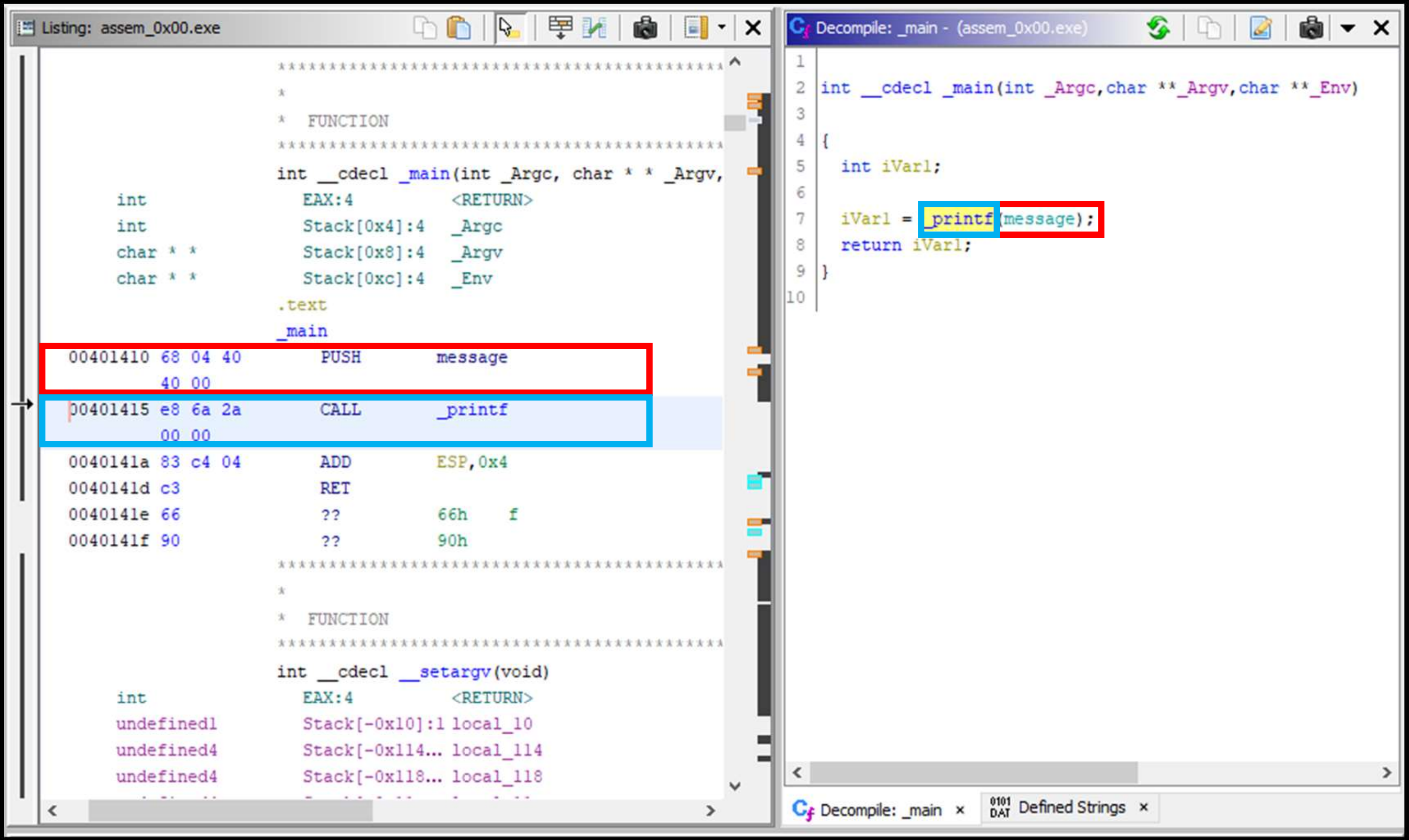
```
                          ******************************************
                          *
                          *  FUNCTION
                          ******************************************
                          int __cdecl _main(int _Argc, char * * _Argv,
      int                   EAX:4          <RETURN>
      int                   Stack[0x4]:4   _Argc
      char * *              Stack[0x8]:4   _Argv
      char * *              Stack[0xc]:4   _Env
                          .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
```

```
int __cdecl _main(int _Argc,char **_Argv,char **_Env)

{
  int iVar1;

  iVar1 = _printf(message);
  return iVar1;
}
```

Assembly
vs
Decompiler

Listing (ASM):
- push message
- call _printf

Decompiler (C)
- call _printf
- pass message



```
                              ******************************************
                              *
                              *  FUNCTION
                              ******************************************
                              int __cdecl _main(int _Argc, char * * _Argv,
          int                   EAX:4          <RETURN>
          int                   Stack[0x4]:4   _Argc
          char * *              Stack[0x8]:4   _Argv
          char * *              Stack[0xc]:4   _Env
                              .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
```

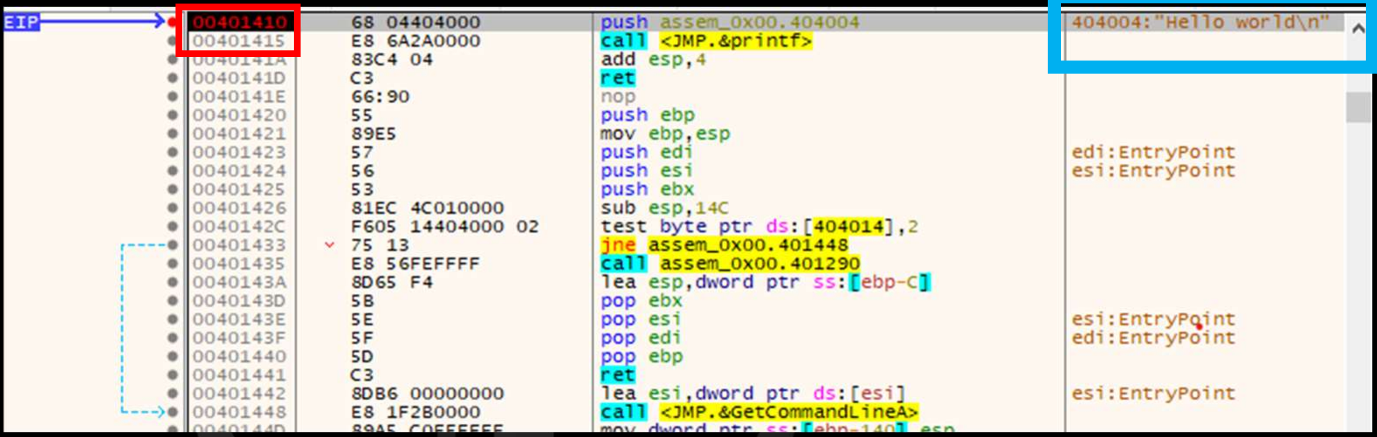
```

int __cdecl _main(int _Argc,char **_Argv,char **_Env)

{
  int iVar1;

  iVar1 = _printf(message);
  return iVar1;
}

```

x64dbg
Ghidra
Workflow

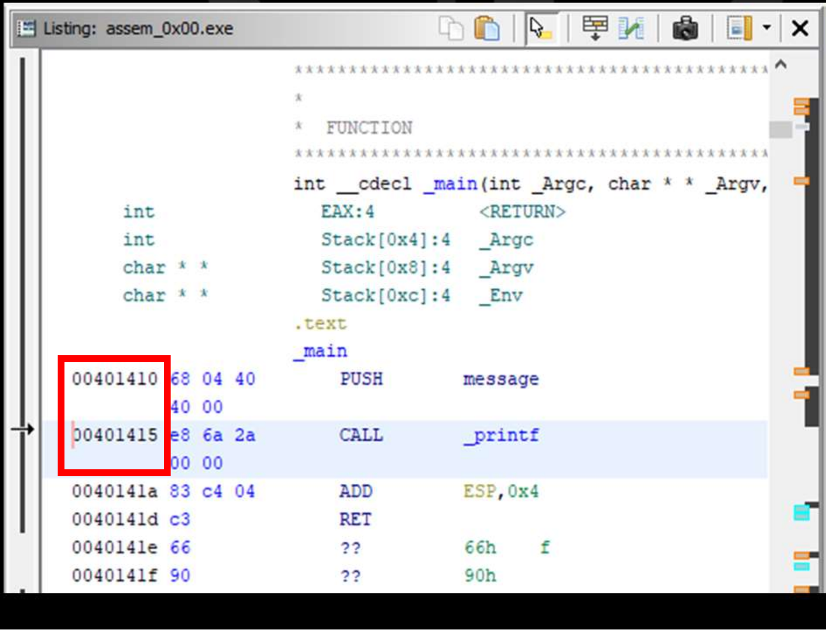
Workflow

x64dbg
Dynamic
00401410
“Hello World”

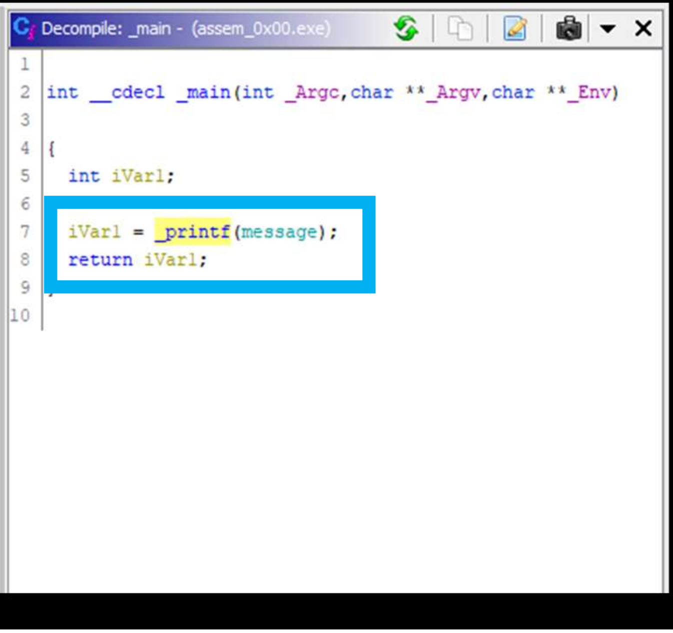
Ghidra
Static
00401410
(message)



```
EIP → 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
```



```
                 ********************************************
                 *
                 *  FUNCTION
                 ********************************************
                 int __cdecl _main(int _Argc, char * * _Argv,
     int           EAX:4          <RETURN>
     int           Stack[0x4]:4   _Argc
     char * *      Stack[0x8]:4   _Argv
     char * *      Stack[0xc]:4   _Env
                 .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
```



```
int __cdecl _main(int _Argc,char **_Argv,char **_Env)

{
  int iVar1;

  iVar1 = _printf(message);
  return iVar1;
}
```

DEMO

Questions?

Jack Ward

Black Lantern Security

jack.ward@blsgvt.com