DLL Injection and x86 Hooking Demystified

Giorgio Gori

Sources:

What is a DLL?
https://support.microsoft.com/en-ca/kb/815065

Windows DLL Injection Basics by Brad Antoniewicz
http://blog.opensecurityresearch.com/2013/01/windows-dll-injection-basics.html

x86 API Hooking Demystified by Jurriaan Bremer
http://jbremer.org/x86-api-hooking-demystified/

What is a DLL?

A DLL - Dynamic Link Library - is a library that contains code and data that can be used by more than one program at the same time.

- Uses fewer resources
- Promotes modular architecture
- Eases deployment and installation

Creating a DLL

```
BOOL APIENTRY DllMain(HANDLE hModule,
    DWORD ul_reason_for_call, LPVOID lpReserved ) {

    switch ( ul_reason_for_call ) {
        case DLL_PROCESS_ATTACHED: // A process is loading the DLL.
        case DLL_THREAD_ATTACHED:  // A process is creating a new thread.
        case DLL_THREAD_DETACH:    // A thread exits normally.
        case DLL_PROCESS_DETACH:   // A process unloads the DLL.
        break;
    }

    return TRUE;
}


extern __declspec(dllexport) void HelloWorld() {
   MessageBox( NULL, TEXT("Hello World"), TEXT("In a DLL"), MB_OK);
}
```

Using a DLL

- Load-time dynamic linking
  Provide a header (.h) and library (.lib) at compile and link time. Linker will provide information to resolve the DLL functions at load time.

```
#include "MyDLL.h"

int main() {
    HelloWorld();
    return 0;
}
```

Using a DLL

- Run-time dynamic linking
  Call LoadLibrary(...) and GetProcAddress(...) at run time, then call the function by address.

```c
int main() {
    HMODULE dll = LoadLibrary("MyDLL.dll");
    if (dll != NULL) {
        FARPROC HelloWorld = GetProcAddress(dll, "HelloWorld");
        if (HelloWorld != NULL)
            HelloWorld();

        FreeLibrary(dll);
    }
    return 0;
}
```

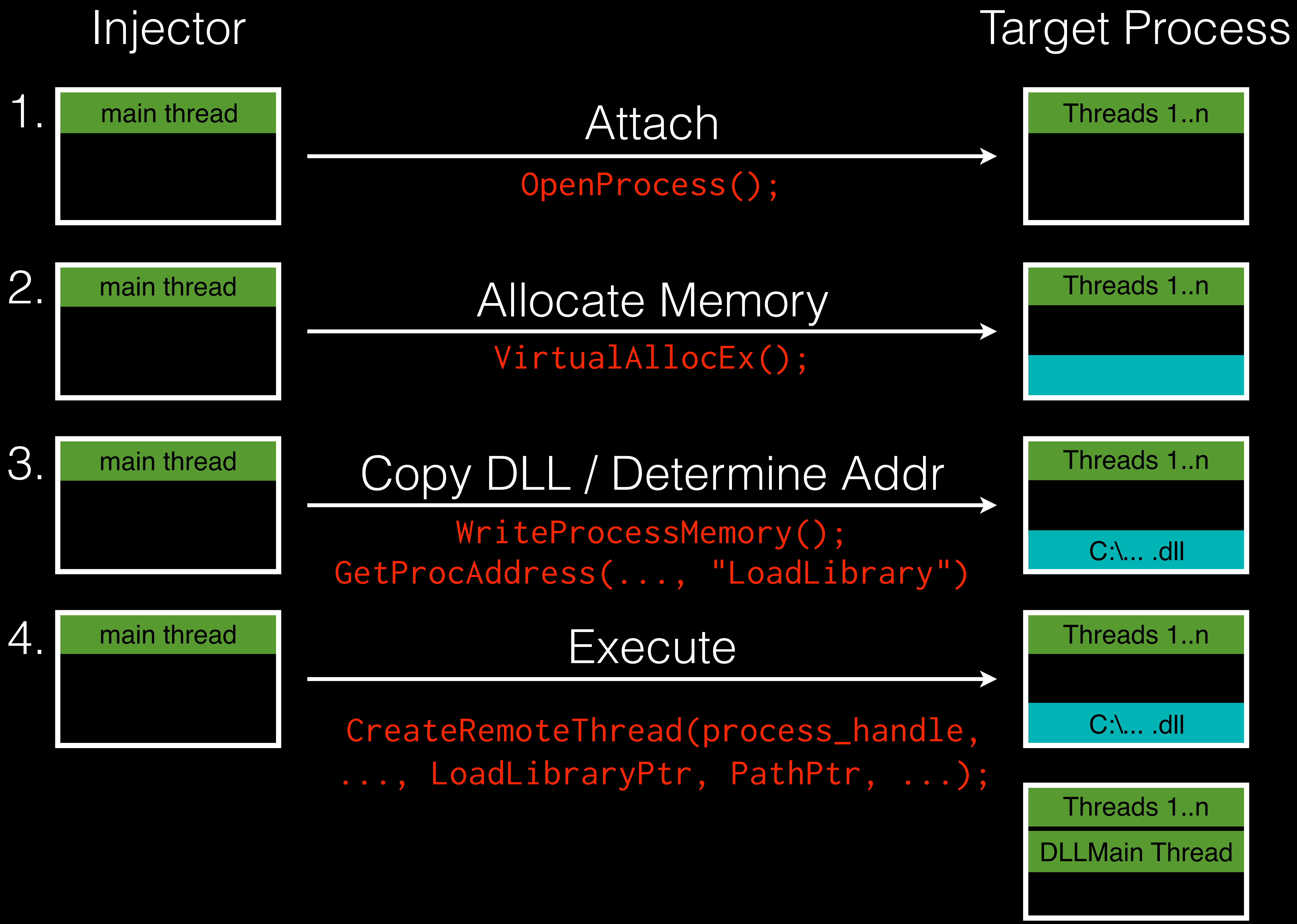
DLL Injection

Invoke `LoadLibrary` **from** the target process

Create a Thread, use `LoadLibrary` as entry point, and the dll path as argument

DLL Injection

1. Attach to the target process.
2. Allocate memory within the process.
3. Copy DLL path into the process memory and find LoadLibrary address.
4. Execute your DLL.



```
OpenProcess();
```

```
VirtualAllocEx();
```

```
WriteProcessMemory();
GetProcAddress(..., "LoadLibrary")
```

```
CreateRemoteThread(process_handle,
..., LoadLibraryPtr, PathPtr, ...);
```

DLL Proxying, DLL Hijacking

- Both work by impersonating the legitimate DLL and (typically) relaying functionality to it. They can be used both to extend functionality and as a malicious attack vector.
- Proxying: Rename the legitimate DLL, replace with your own.
- Hijacking: Abuse Windows' DLL Search order to load your DLL before the legitimate one.

DLL Injection: Why?

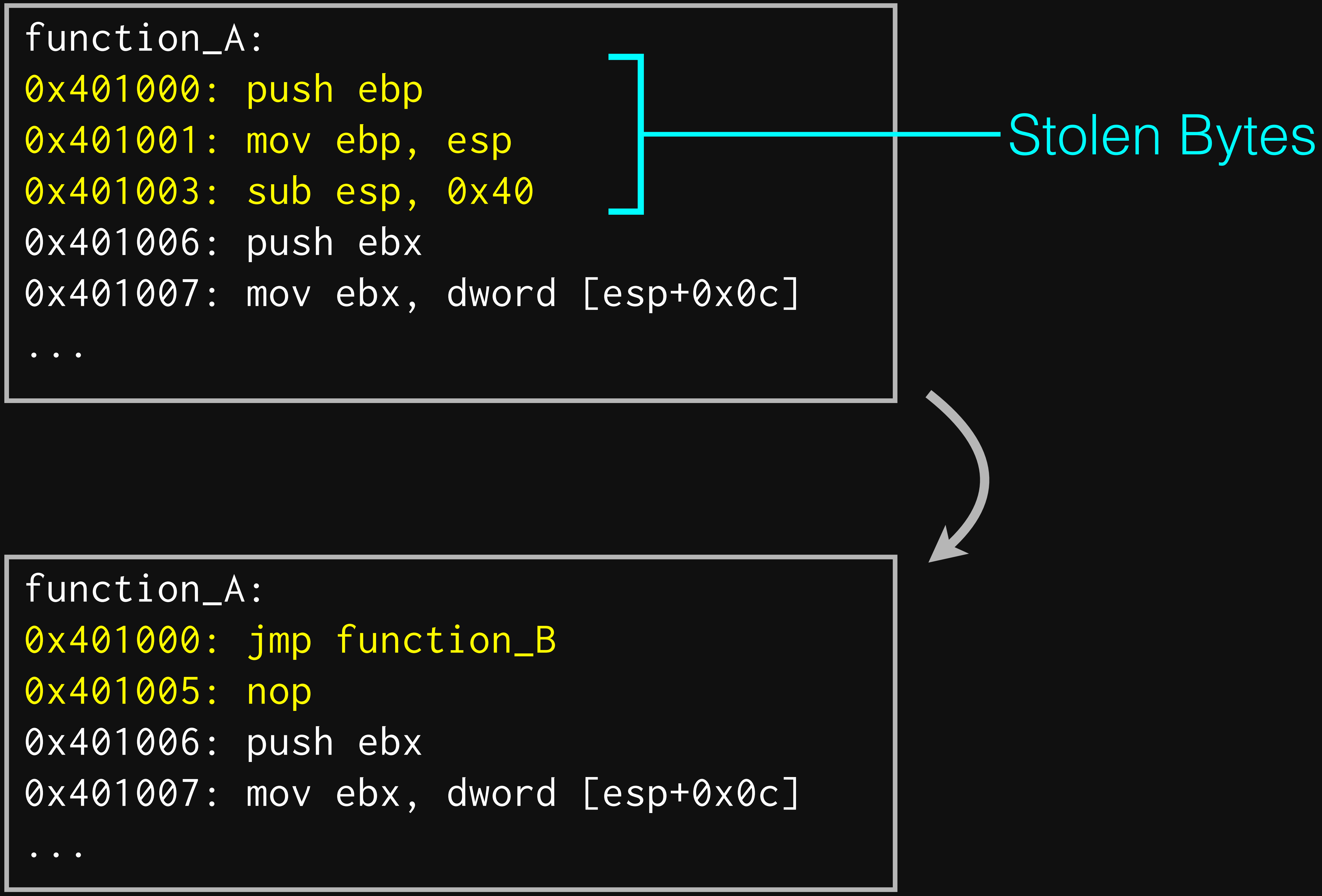
- Read and write process memory
- Execute custom code, invoke existing functions
- Patch binary code, add hooks

x86 Hooking

Change the byte code to alter the execution.
Common uses include:

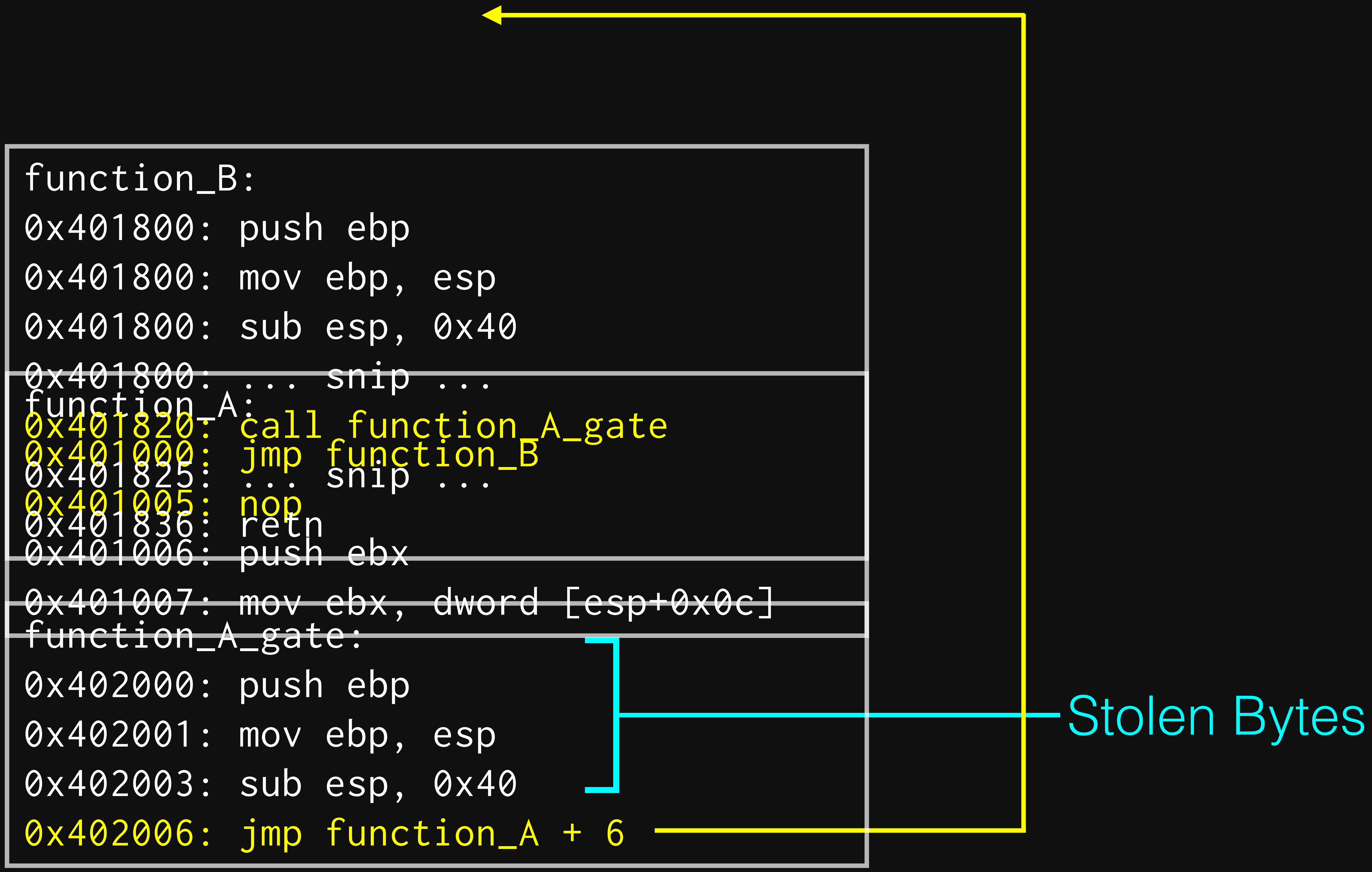
- Debugging.
- Profiling.
- Extending functionality.
- Execute general "on event" code.

```
function_A:
0x401000: push ebp
0x401001: mov ebp, esp
0x401003: sub esp, 0x40
0x401006: push ebx
0x401007: mov ebx, dword [esp+0x0c]
...
```



```
function_A:
0x401000: push ebp
0x401001: mov ebp, esp
0x401003: sub esp, 0x40
0x401006: push ebx
0x401007: mov ebx, dword [esp+0x0c]
...
```

```
function_A:
0x401000: jmp function_B
0x401005: nop
0x401006: push ebx
0x401007: mov ebx, dword [esp+0x0c]
...
```



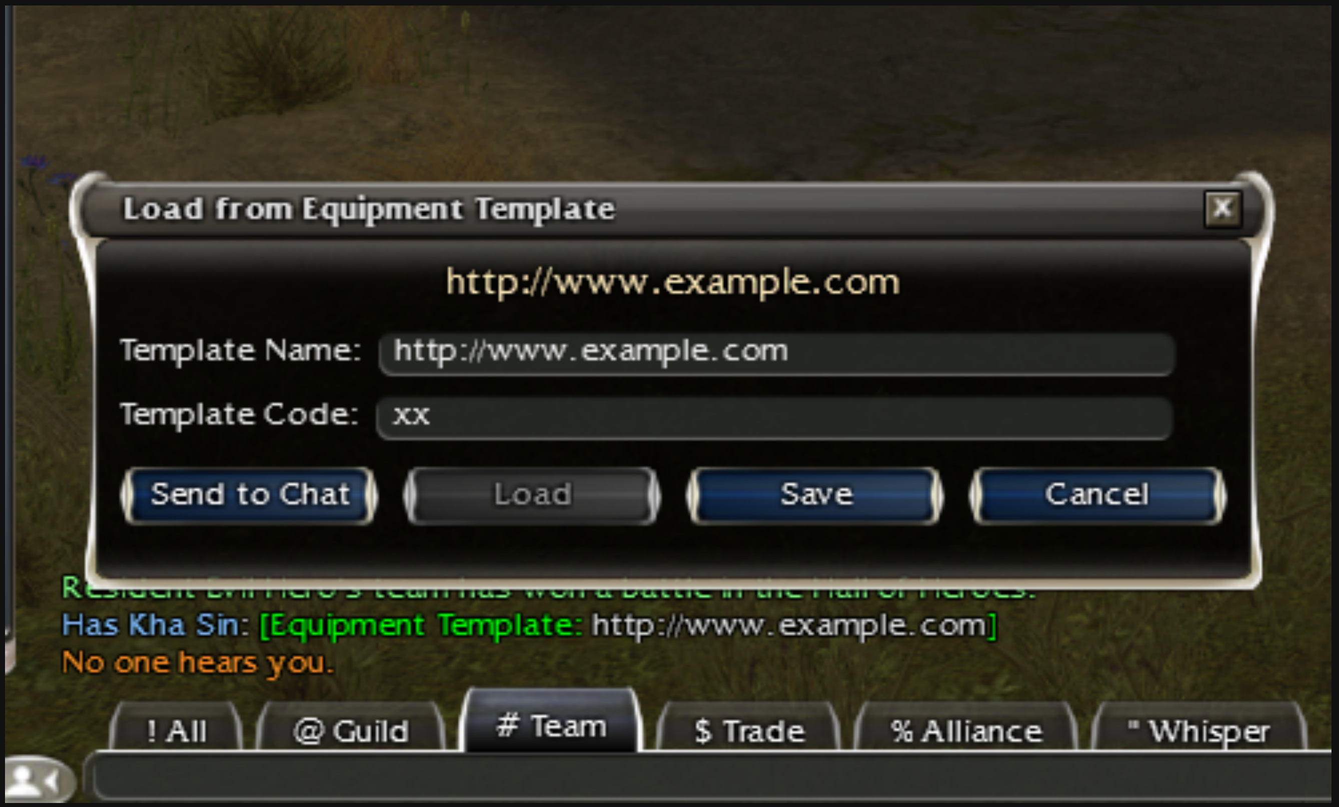
```
function_B:
0x401800: push ebp
0x401800: mov ebp, esp
0x401800: sub esp, 0x40
0x401800: ... snip ...
0x401820: call function_A_gate
0x401825: ... snip ...
0x401836: retn
```

```
function_A:
0x401000: jmp function_B
0x401005: nop
0x401006: push ebx
0x401007: mov ebx, dword [esp+0x0c]
```

```
function_A_gate:
0x402000: push ebp
0x402001: mov ebp, esp
0x402003: sub esp, 0x40
0x402006: jmp function_A + 6
```

Hooking example

- Game does not support clickable links. Players have to click, select, copy, paste in web browser.
- We follow the call from the input handler to the UI creation.
- Hook the function that creates the UI element.
- Open in web browser if the name is a URL.


Load from Equipment Template

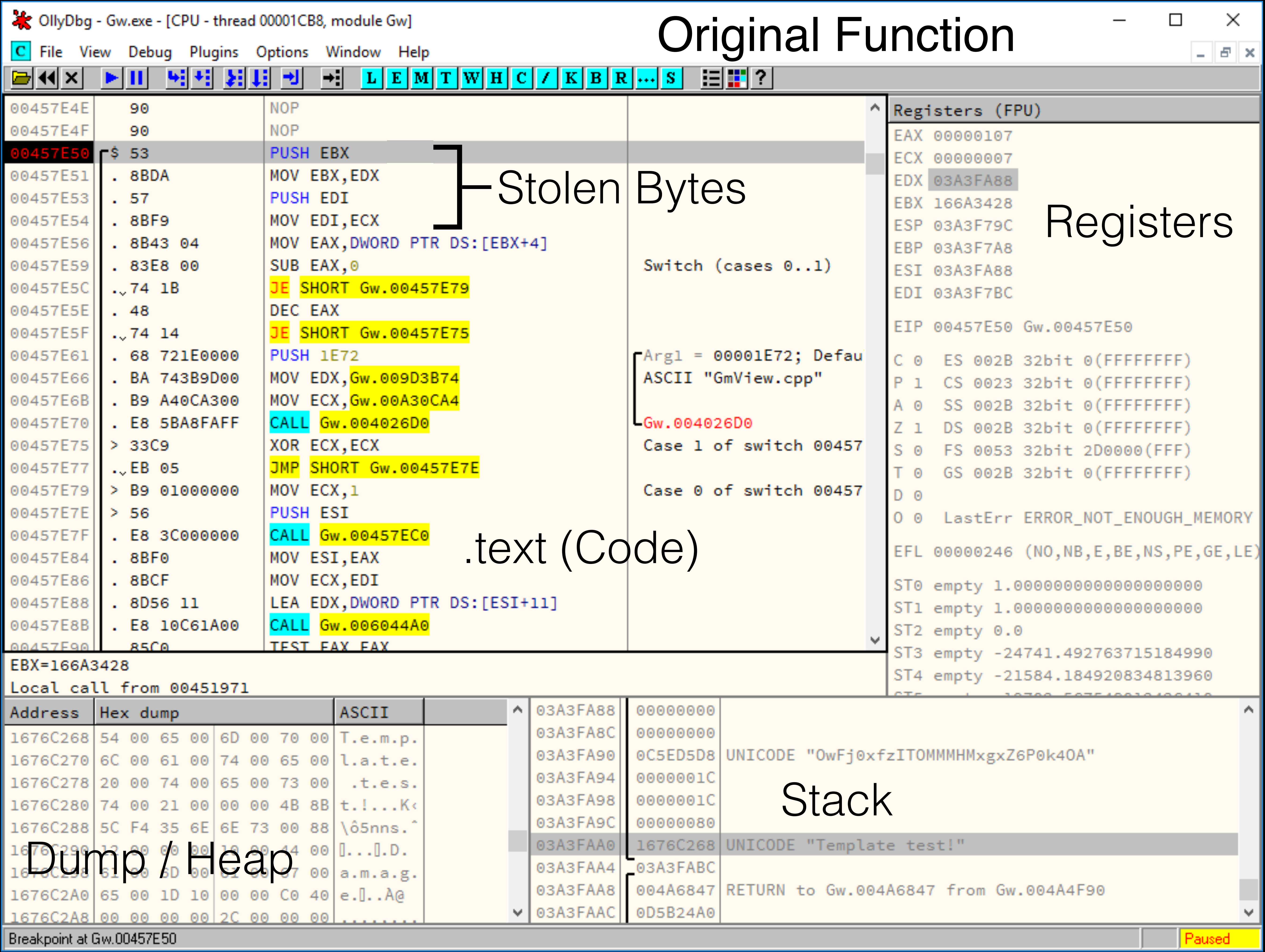
http://www.example.com

Template Name: http://www.example.com

Template Code: xx

Has Kha Sin: [Equipment Template: http://www.example.com]

No one hears you.


OllyDbg - Gw.exe - [CPU - thread 00001CB8, module Gw]
Original Function
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
Stolen Bytes
Registers
.text (Code)
Stack
Dump / Heap
Registers (FPU)
Address Hex dump ASCII
Paused

```
00457E4E      90              NOP
00457E4F      90              NOP
00457E50   ┌$ 53              PUSH EBX
00457E51   │. 8BDA            MOV EBX,EDX
00457E53   │. 57              PUSH EDI
00457E54   │. 8BF9            MOV EDI,ECX
00457E56   │. 8B43 04         MOV EAX,DWORD PTR DS:[EBX+4]
00457E59   │. 83E8 00         SUB EAX,0                            Switch (cases 0..1)
00457E5C   │.v74 1B           JE SHORT Gw.00457E79
00457E5E   │. 48              DEC EAX
00457E5F   │.v74 14           JE SHORT Gw.00457E75
00457E61   │. 68 721E0000     PUSH 1E72                           ┌Arg1 = 00001E72; Defau
00457E66   │. BA 743B9D00     MOV EDX,Gw.009D3B74                 │ASCII "GmView.cpp"
00457E6B   │. B9 A40CA300     MOV ECX,Gw.00A30CA4                 │
00457E70   │. E8 5BA8FAFF     CALL Gw.004026D0                    └Gw.004026D0
00457E75   │> 33C9            XOR ECX,ECX                          Case 1 of switch 00457
00457E77   │.v EB 05          JMP SHORT Gw.00457E7E
00457E79   │> B9 01000000     MOV ECX,1                            Case 0 of switch 00457
00457E7E   │> 56              PUSH ESI
00457E7F   │. E8 3C000000     CALL Gw.00457EC0
00457E84   │. 8BF0            MOV ESI,EAX
00457E86   │. 8BCF            MOV ECX,EDI
00457E88   │. 8D56 11         LEA EDX,DWORD PTR DS:[ESI+11]
00457E8B   │. E8 10C61A00     CALL Gw.006044A0
00457E90   │. 85C0            TEST EAX,EAX
```

```
EBX=166A3428
Local call from 00451971
```

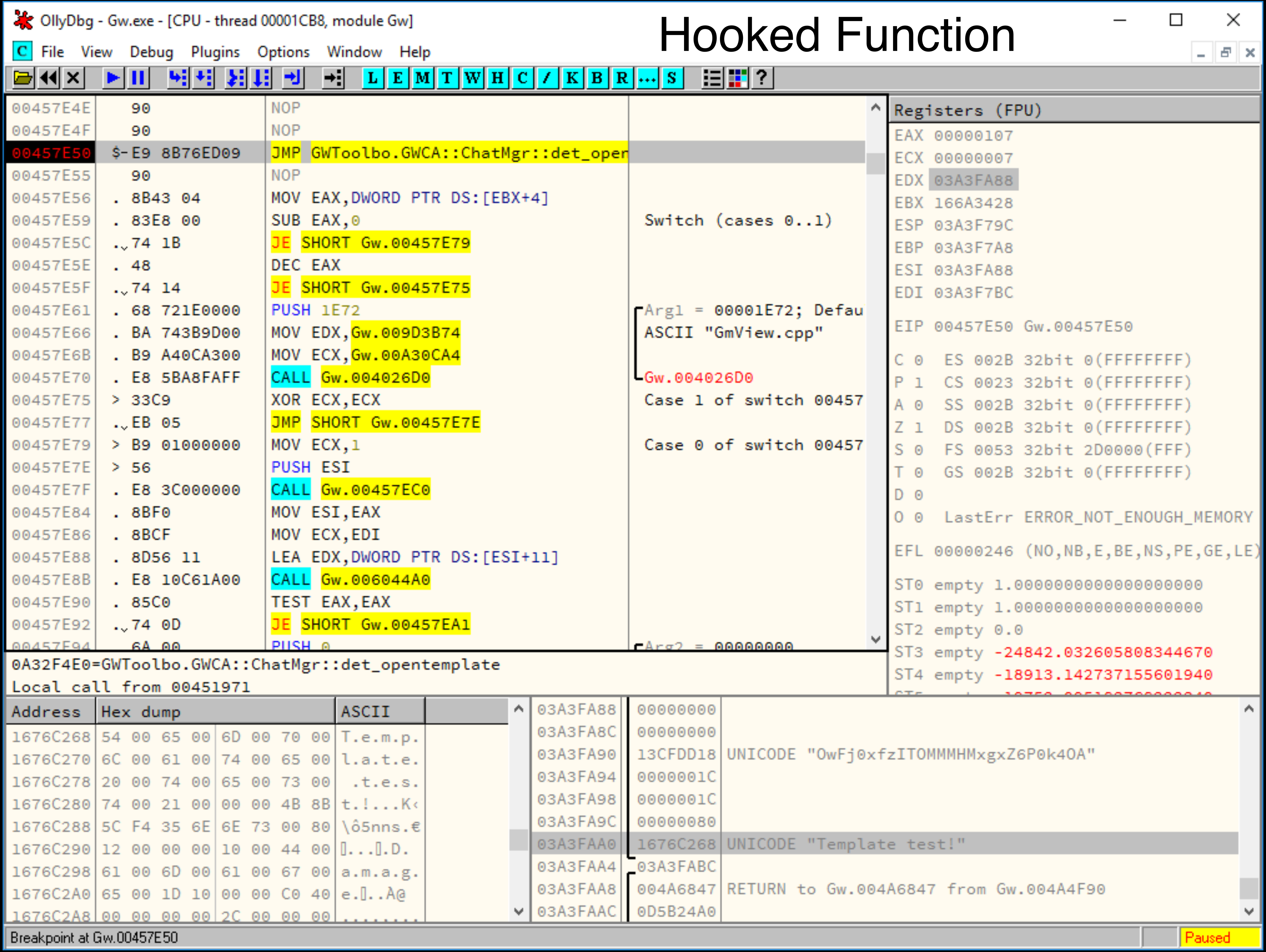
```
Registers (FPU)
EAX 00000107
ECX 00000007
EDX 03A3FA88
EBX 166A3428
ESP 03A3F79C
EBP 03A3F7A8
ESI 03A3FA88
EDI 03A3F7BC

EIP 00457E50 Gw.00457E50

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 2D0000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_NOT_ENOUGH_MEMORY

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty 1.0000000000000000000
ST1 empty 1.0000000000000000000
ST2 empty 0.0
ST3 empty -24741.492763715184990
ST4 empty -21584.184920834813960
```

```
Address   Hex dump                  ASCII
1676C268  54 00 65 00 6D 00 70 00  T.e.m.p.
1676C270  6C 00 61 00 74 00 65 00  l.a.t.e.
1676C278  20 00 74 00 65 00 73 00   .t.e.s.
1676C280  74 00 21 00 00 00 4B 8B  t.!...K‹
1676C288  5C F4 35 6E 6E 73 00 88  \ô5nns.ˆ
1676C290  12 00 00 00 10 00 44 00  ▯...▯.D.
1676C298  61 00 6D 00 61 00 67 00  a.m.a.g.
1676C2A0  65 00 1D 10 00 00 C0 40  e.▯..À@
1676C2A8  00 00 00 00 2C 00 00 00  ........
```

```
03A3FA88   00000000
03A3FA8C   00000000
03A3FA90   0C5ED5D8  UNICODE "OwFj0xfzITOMMMHMxgxZ6P0k4OA"
03A3FA94   0000001C
03A3FA98   0000001C
03A3FA9C   00000080
03A3FAA0   1676C268  UNICODE "Template test!"
03A3FAA4   03A3FABC
03A3FAA8   004A6847  RETURN to Gw.004A6847 from Gw.004A4F90
03A3FAAC   0D5B24A0
```

```
Breakpoint at Gw.00457E50
```

Hooked Function


OllyDbg - Gw.exe - [CPU - thread 00001CB8, module Gw]
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S

```
00457E4E    90          NOP
00457E4F    90          NOP
00457E50  $-E9 8B76ED09 JMP GWToolbo.GWCA::ChatMgr::det_open
00457E55    90          NOP
00457E56  . 8B43 04     MOV EAX,DWORD PTR DS:[EBX+4]
00457E59  . 83E8 00     SUB EAX,0                        Switch (cases 0..1)
00457E5C  .v74 1B       JE SHORT Gw.00457E79
00457E5E  . 48          DEC EAX
00457E5F  .v74 14       JE SHORT Gw.00457E75
00457E61  . 68 721E0000 PUSH 1E72                        Arg1 = 00001E72; Defau
00457E66  . BA 743B9D00 MOV EDX,Gw.009D3B74              ASCII "GmView.cpp"
00457E6B  . B9 A40CA300 MOV ECX,Gw.00A30CA4
00457E70  . E8 5BA8FAFF CALL Gw.004026D0                 Gw.004026D0
00457E75  > 33C9        XOR ECX,ECX                      Case 1 of switch 00457
00457E77  .vEB 05       JMP SHORT Gw.00457E7E
00457E79  > B9 01000000 MOV ECX,1                        Case 0 of switch 00457
00457E7E  > 56          PUSH ESI
00457E7F  . E8 3C000000 CALL Gw.00457EC0
00457E84  . 8BF0        MOV ESI,EAX
00457E86  . 8BCF        MOV ECX,EDI
00457E88  . 8D56 11     LEA EDX,DWORD PTR DS:[ESI+11]
00457E8B  . E8 10C61A00 CALL Gw.006044A0
00457E90  . 85C0        TEST EAX,EAX
00457E92  .v74 0D       JE SHORT Gw.00457EA1
00457E94    6A 00       PUSH 0                           Arg2 = 00000000
```

```
0A32F4E0=GWToolbo.GWCA::ChatMgr::det_opentemplate
Local call from 00451971
```

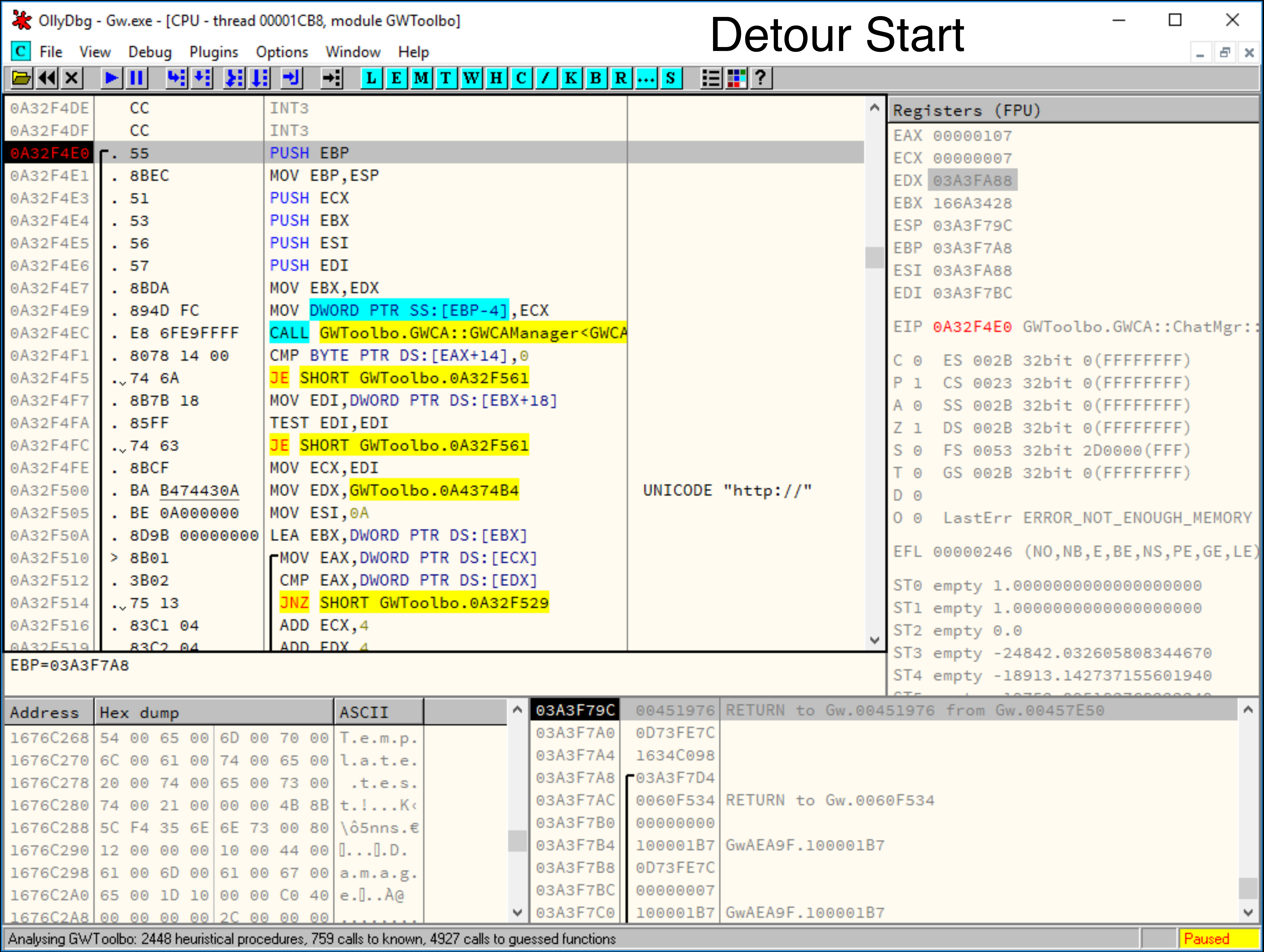
```
Registers (FPU)
EAX 00000107
ECX 00000007
EDX 03A3FA88
EBX 166A3428
ESP 03A3F79C
EBP 03A3F7A8
ESI 03A3FA88
EDI 03A3F7BC

EIP 00457E50 Gw.00457E50

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 2D0000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_NOT_ENOUGH_MEMORY

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty 1.0000000000000000000
ST1 empty 1.0000000000000000000
ST2 empty 0.0
ST3 empty -24842.032605808344670
ST4 empty -18913.142737155601940
```

```
Address  Hex dump                 ASCII
1676C268 54 00 65 00 6D 00 70 00 T.e.m.p.
1676C270 6C 00 61 00 74 00 65 00 l.a.t.e.
1676C278 20 00 74 00 65 00 73 00  .t.e.s.
1676C280 74 00 21 00 00 00 4B 8B t.!...K‹
1676C288 5C F4 35 6E 6E 73 00 80 \ô5nns.€
1676C290 12 00 00 00 10 00 44 00 ....D.
1676C298 61 00 6D 00 61 00 67 00 a.m.a.g.
1676C2A0 65 00 1D 10 00 00 C0 40 e.....À@
1676C2A8 00 00 00 00 2C 00 00 00 ........
```

```
03A3FA88  00000000
03A3FA8C  00000000
03A3FA90  13CFDD18  UNICODE "OwFj0xfzITOMMMHMxgxZ6P0k40A"
03A3FA94  0000001C
03A3FA98  0000001C
03A3FA9C  00000080
03A3FAA0  1676C268  UNICODE "Template test!"
03A3FAA4  03A3FABC
03A3FAA8  004A6847  RETURN to Gw.004A6847 from Gw.004A4F90
03A3FAAC  0D5B24A0
```

```
Breakpoint at Gw.00457E50                                        Paused
```

Detour Start



```
OllyDbg - Gw.exe - [CPU - thread 00001CB8, module GWToolbo]
```

File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S

```
0A32F4DE     CC            INT3
0A32F4DF     CC            INT3
0A32F4E0   ┌. 55            PUSH EBP
0A32F4E1   . 8BEC          MOV EBP,ESP
0A32F4E3   . 51            PUSH ECX
0A32F4E4   . 53            PUSH EBX
0A32F4E5   . 56            PUSH ESI
0A32F4E6   . 57            PUSH EDI
0A32F4E7   . 8BDA          MOV EBX,EDX
0A32F4E9   . 894D FC       MOV DWORD PTR SS:[EBP-4],ECX
0A32F4EC   . E8 6FE9FFFF   CALL GWToolbo.GWCA::GWCAManager<GWCA
0A32F4F1   . 8078 14 00    CMP BYTE PTR DS:[EAX+14],0
0A32F4F5   .v 74 6A        JE SHORT GWToolbo.0A32F561
0A32F4F7   . 8B7B 18       MOV EDI,DWORD PTR DS:[EBX+18]
0A32F4FA   . 85FF          TEST EDI,EDI
0A32F4FC   .v 74 63        JE SHORT GWToolbo.0A32F561
0A32F4FE   . 8BCF          MOV ECX,EDI
0A32F500   . BA B474430A   MOV EDX,GWToolbo.0A4374B4         UNICODE "http://"
0A32F505   . BE 0A000000   MOV ESI,0A
0A32F50A   . 8D9B 00000000 LEA EBX,DWORD PTR DS:[EBX]
0A32F510   > 8B01          ┌MOV EAX,DWORD PTR DS:[ECX]
0A32F512   . 3B02          │CMP EAX,DWORD PTR DS:[EDX]
0A32F514   .v 75 13        │JNZ SHORT GWToolbo.0A32F529
0A32F516   . 83C1 04       │ADD ECX,4
0A32F519   . 83C2 04       │ADD EDX,4
EBP=03A3F7A8
```

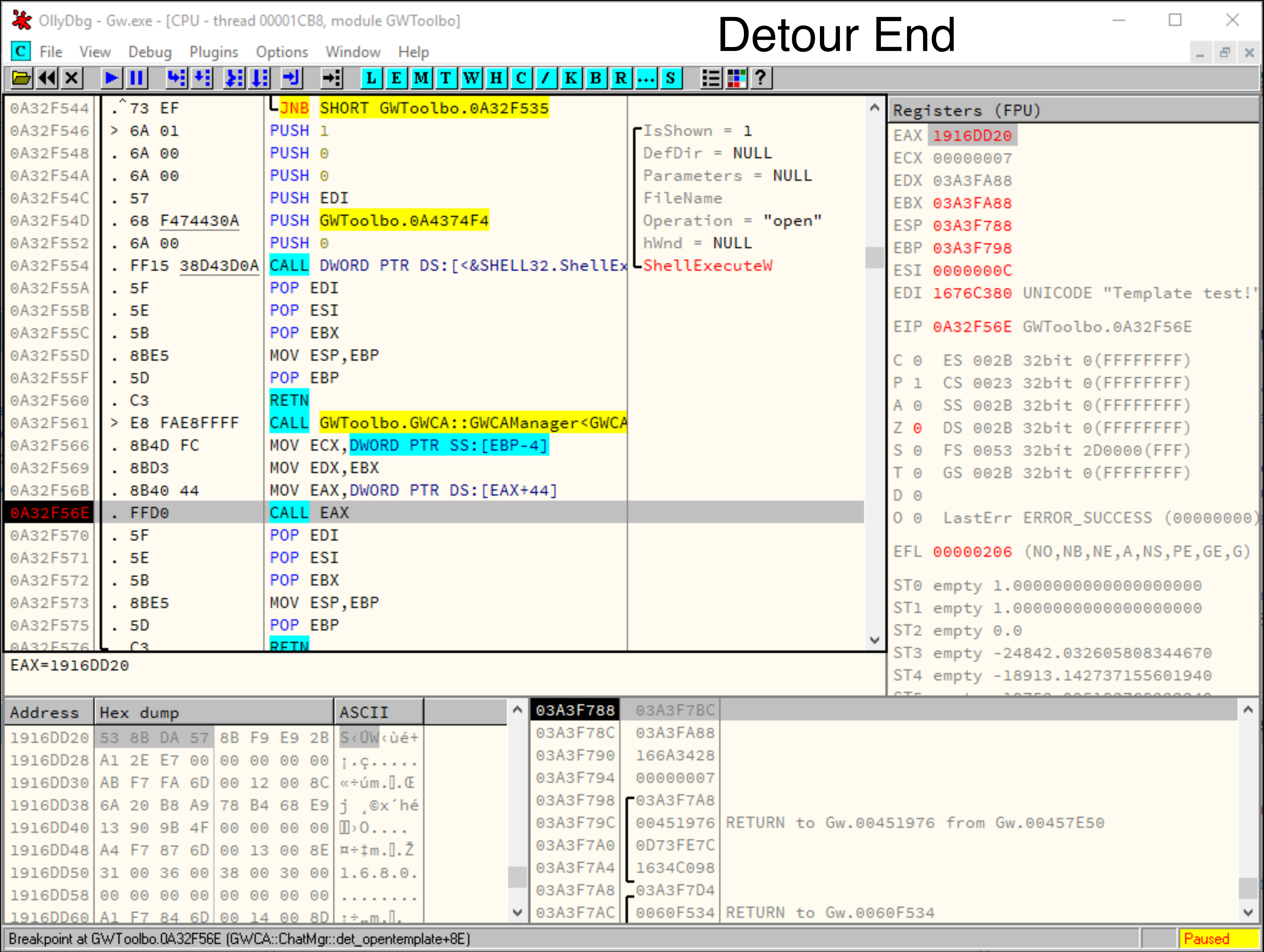
```
Registers (FPU)
EAX 00000107
ECX 00000007
EDX 03A3FA88
EBX 166A3428
ESP 03A3F79C
EBP 03A3F7A8
ESI 03A3FA88
EDI 03A3F7BC

EIP 0A32F4E0 GWToolbo.GWCA::ChatMgr::

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 2D0000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_NOT_ENOUGH_MEMORY

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty 1.0000000000000000000
ST1 empty 1.0000000000000000000
ST2 empty 0.0
ST3 empty -24842.032605808344670
ST4 empty -18913.142737155601940
```

```
Address   Hex dump                  ASCII
1676C268  54 00 65 00 6D 00 70 00   T.e.m.p.
1676C270  6C 00 61 00 74 00 65 00   l.a.t.e.
1676C278  20 00 74 00 65 00 73 00    .t.e.s.
1676C280  74 00 21 00 00 00 4B 8B   t.!...K‹
1676C288  5C F4 35 6E 6E 73 00 80   \ô5nns.€
1676C290  12 00 00 00 10 00 44 00   ▯...▯.D.
1676C298  61 00 6D 00 61 00 67 00   a.m.a.g.
1676C2A0  65 00 1D 10 00 00 C0 40   e.▯..À@
1676C2A8  00 00 00 00 2C 00 00 00   ........
```

```
03A3F79C  00451976  RETURN to Gw.00451976 from Gw.00457E50
03A3F7A0  0D73FE7C
03A3F7A4  1634C098
03A3F7A8 ┌03A3F7D4
03A3F7AC │0060F534  RETURN to Gw.0060F534
03A3F7B0 │00000000
03A3F7B4 │100001B7  GwAEA9F.100001B7
03A3F7B8 │0D73FE7C
03A3F7BC │00000007
03A3F7C0 │100001B7  GwAEA9F.100001B7
```

```
Analysing GWToolbo: 2448 heuristical procedures, 759 calls to known, 4927 calls to guessed functions    Paused
```

Detour End


OllyDbg - Gw.exe - [CPU - thread 00001CB8, module GWToolbo]
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
Registers (FPU)
Address Hex dump ASCII
Paused

```
0A32F544  .^73 EF          └JNB SHORT GWToolbo.0A32F535
0A32F546  > 6A 01           PUSH 1                                  ┌IsShown = 1
0A32F548  . 6A 00           PUSH 0                                  │DefDir = NULL
0A32F54A  . 6A 00           PUSH 0                                  │Parameters = NULL
0A32F54C  . 57              PUSH EDI                                │FileName
0A32F54D  . 68 F474430A     PUSH GWToolbo.0A4374F4                  │Operation = "open"
0A32F552  . 6A 00           PUSH 0                                  │hWnd = NULL
0A32F554  . FF15 38D43D0A   CALL DWORD PTR DS:[<&SHELL32.ShellEx    └ShellExecuteW
0A32F55A  . 5F              POP EDI
0A32F55B  . 5E              POP ESI
0A32F55C  . 5B              POP EBX
0A32F55D  . 8BE5            MOV ESP,EBP
0A32F55F  . 5D              POP EBP
0A32F560  . C3              RETN
0A32F561  > E8 FAE8FFFF     CALL GWToolbo.GWCA::GWCAManager<GWCA
0A32F566  . 8B4D FC         MOV ECX,DWORD PTR SS:[EBP-4]
0A32F569  . 8BD3            MOV EDX,EBX
0A32F56B  . 8B40 44         MOV EAX,DWORD PTR DS:[EAX+44]
0A32F56E  . FFD0            CALL EAX
0A32F570  . 5F              POP EDI
0A32F571  . 5E              POP ESI
0A32F572  . 5B              POP EBX
0A32F573  . 8BE5            MOV ESP,EBP
0A32F575  . 5D              POP EBP
0A32F576    C3              RETN
EAX=1916DD20
```

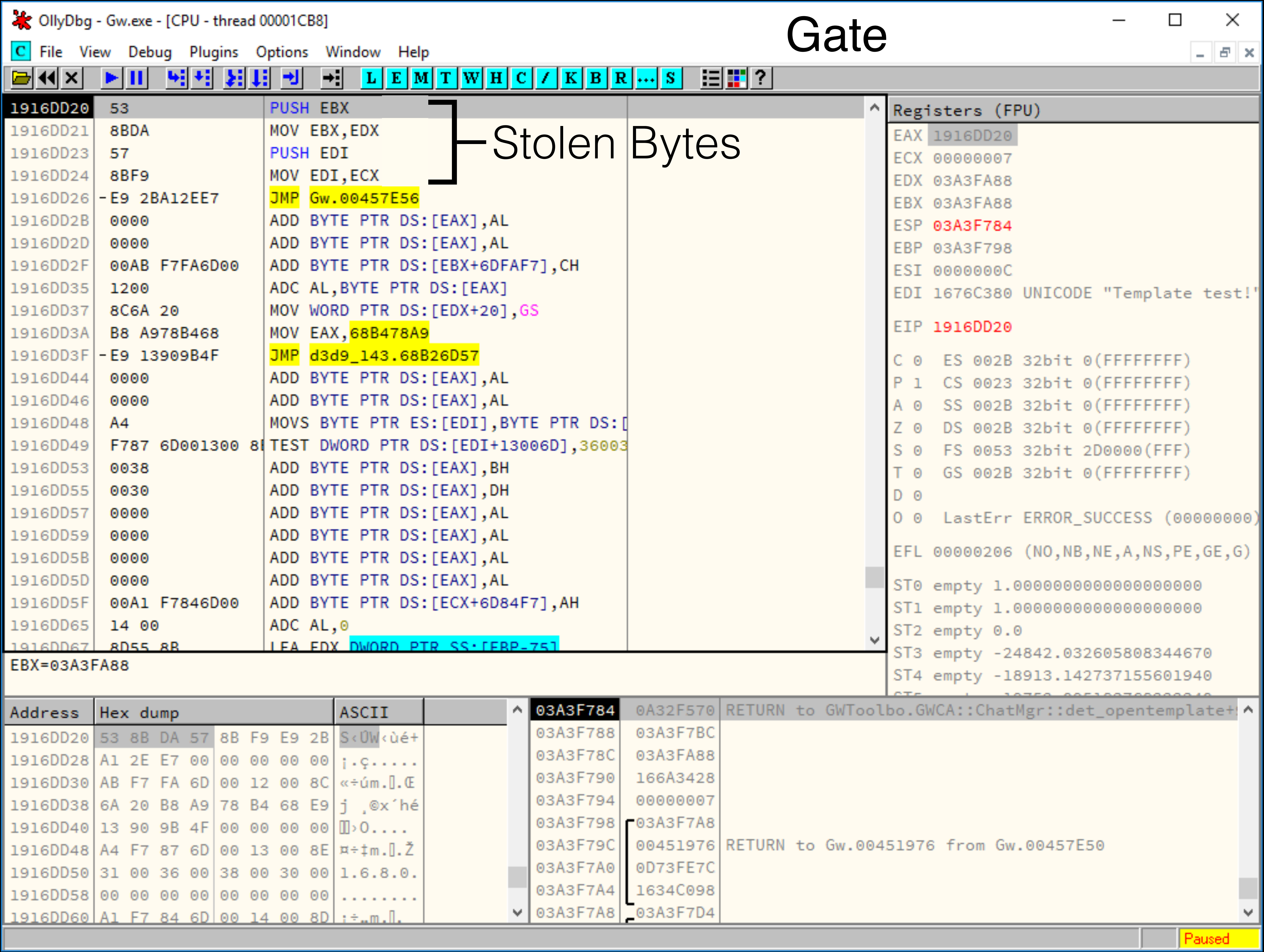
```
EAX 1916DD20
ECX 00000007
EDX 03A3FA88
EBX 03A3FA88
ESP 03A3F788
EBP 03A3F798
ESI 0000000C
EDI 1676C380 UNICODE "Template test!"

EIP 0A32F56E GWToolbo.0A32F56E

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 2D0000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

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

ST0 empty 1.0000000000000000000
ST1 empty 1.0000000000000000000
ST2 empty 0.0
ST3 empty -24842.032605808344670
ST4 empty -18913.142737155601940
```

```
Address   Hex dump                  ASCII
1916DD20  53 8B DA 57 8B F9 E9 2B   S‹ÚW‹ùé+
1916DD28  A1 2E E7 00 00 00 00 00   ¡.ç.....
1916DD30  AB F7 FA 6D 00 12 00 8C   «÷úm.].Œ
1916DD38  6A 20 B8 A9 78 B4 68 E9   j ¸©x´hé
1916DD40  13 90 9B 4F 00 00 00 00   ]]›O....
1916DD48  A4 F7 87 6D 00 13 00 8E   ¤÷‡m.].Ž
1916DD50  31 00 36 00 38 00 30 00   1.6.8.0.
1916DD58  00 00 00 00 00 00 00 00   ........
1916DD60  A1 F7 84 6D 00 14 00 8D   ¡÷„m.].
```

```
03A3F788  03A3F7BC
03A3F78C  03A3FA88
03A3F790  166A3428
03A3F794  00000007
03A3F798 ┌03A3F7A8
03A3F79C │00451976  RETURN to Gw.00451976 from Gw.00457E50
03A3F7A0 │0D73FE7C
03A3F7A4 │1634C098
03A3F7A8 ┌03A3F7D4
03A3F7AC │0060F534  RETURN to Gw.0060F534
```

Breakpoint at GWToolbo.0A32F56E (GWCA::ChatMgr::det_opentemplate+8E)

Gate

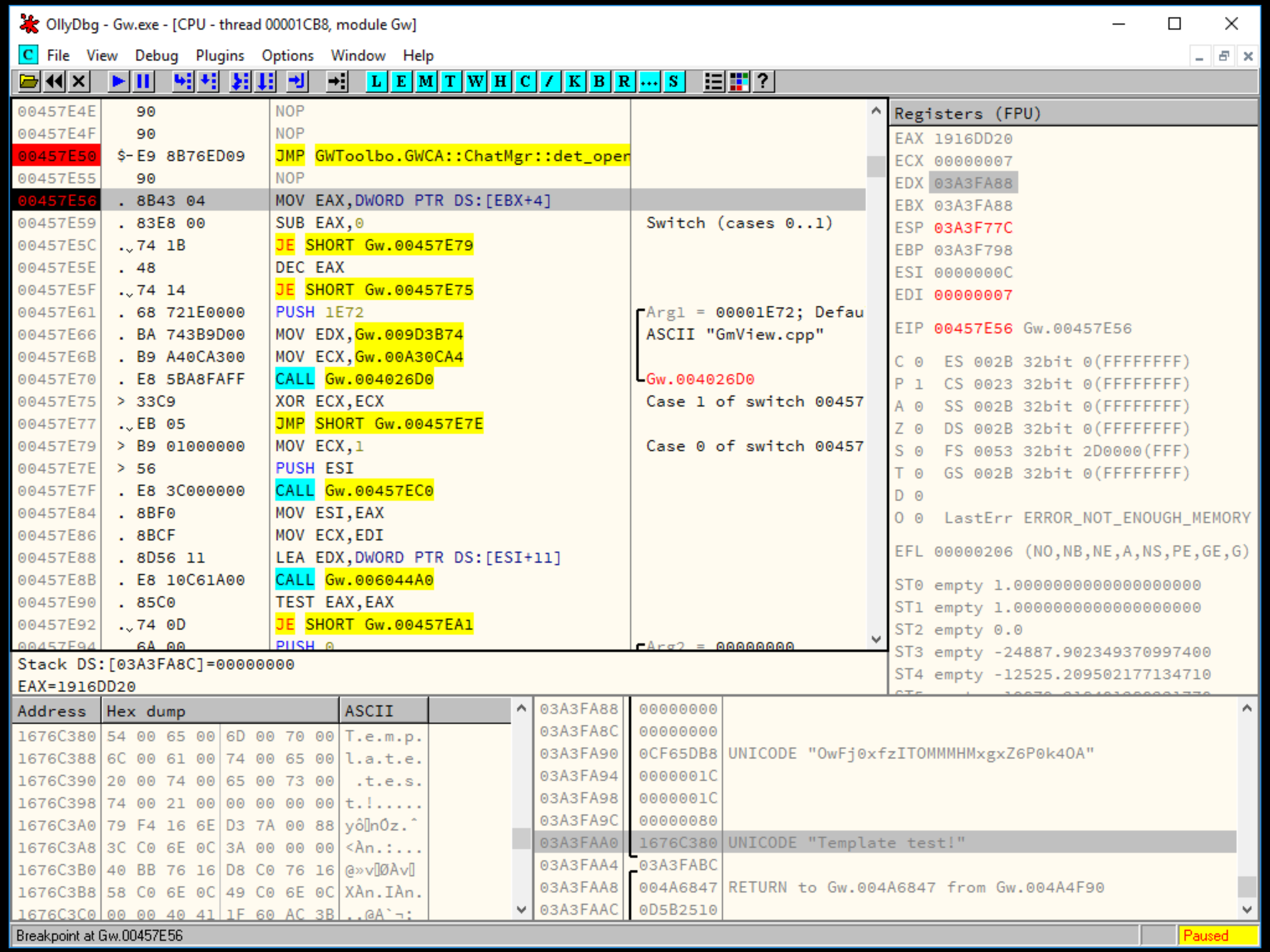

OllyDbg - Gw.exe - [CPU - thread 00001CB8]
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
Stolen Bytes
Paused

```
1916DD20   53                PUSH EBX
1916DD21   8BDA              MOV EBX,EDX
1916DD23   57                PUSH EDI
1916DD24   8BF9              MOV EDI,ECX
1916DD26  -E9 2BA12EE7       JMP Gw.00457E56
1916DD2B   0000              ADD BYTE PTR DS:[EAX],AL
1916DD2D   0000              ADD BYTE PTR DS:[EAX],AL
1916DD2F   00AB F7FA6D00     ADD BYTE PTR DS:[EBX+6DFAF7],CH
1916DD35   1200              ADC AL,BYTE PTR DS:[EAX]
1916DD37   8C6A 20           MOV WORD PTR DS:[EDX+20],GS
1916DD3A   B8 A978B468       MOV EAX,68B478A9
1916DD3F  -E9 13909B4F       JMP d3d9_143.68B26D57
1916DD44   0000              ADD BYTE PTR DS:[EAX],AL
1916DD46   0000              ADD BYTE PTR DS:[EAX],AL
1916DD48   A4                MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[
1916DD49   F787 6D001300 8   TEST DWORD PTR DS:[EDI+13006D],36003
1916DD53   0038              ADD BYTE PTR DS:[EAX],BH
1916DD55   0030              ADD BYTE PTR DS:[EAX],DH
1916DD57   0000              ADD BYTE PTR DS:[EAX],AL
1916DD59   0000              ADD BYTE PTR DS:[EAX],AL
1916DD5B   0000              ADD BYTE PTR DS:[EAX],AL
1916DD5D   0000              ADD BYTE PTR DS:[EAX],AL
1916DD5F   00A1 F7846D00     ADD BYTE PTR DS:[ECX+6D84F7],AH
1916DD65   14 00             ADC AL,0
1916DD67   8D55 8B           LEA EDX,DWORD PTR SS:[EBP-75]
EBX=03A3FA88
```

```
Registers (FPU)
EAX 1916DD20
ECX 00000007
EDX 03A3FA88
EBX 03A3FA88
ESP 03A3F784
EBP 03A3F798
ESI 0000000C
EDI 1676C380 UNICODE "Template test!"

EIP 1916DD20

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 2D0000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

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

ST0 empty 1.0000000000000000000
ST1 empty 1.0000000000000000000
ST2 empty 0.0
ST3 empty -24842.032605808344670
ST4 empty -18913.142737155601940
```

```
Address   Hex dump                 ASCII
1916DD20  53 8B DA 57 8B F9 E9 2B  S‹ÚW‹ùé+
1916DD28  A1 2E E7 00 00 00 00 00  ¡.ç.....
1916DD30  AB F7 FA 6D 00 12 00 8C  «÷úm.].Œ
1916DD38  6A 20 B8 A9 78 B4 68 E9  j ¸©x´hé
1916DD40  13 90 9B 4F 00 00 00 00  ]›O....
1916DD48  A4 F7 87 6D 00 13 00 8E  ¤÷‡m.].Ž
1916DD50  31 00 36 00 38 00 30 00  1.6.8.0.
1916DD58  00 00 00 00 00 00 00 00  ........
1916DD60  A1 F7 84 6D 00 14 00 8D  ¡÷„m.].
```

```
03A3F784   0A32F570  RETURN to GWToolbo.GWCA::ChatMgr::det_opentemplate+
03A3F788   03A3F7BC
03A3F78C   03A3FA88
03A3F790   166A3428
03A3F794   00000007
03A3F798  ┌03A3F7A8
03A3F79C  │00451976  RETURN to Gw.00451976 from Gw.00457E50
03A3F7A0  │0D73FE7C
03A3F7A4  │1634C098
03A3F7A8  └03A3F7D4
```


OllyDbg - Gw.exe - [CPU - thread 00001CB8, module Gw]
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S

```
00457E4E     90              NOP
00457E4F     90              NOP
00457E50   $-E9 8B76ED09     JMP GWToolbo.GWCA::ChatMgr::det_oper
00457E55     90              NOP
00457E56   . 8B43 04         MOV EAX,DWORD PTR DS:[EBX+4]
00457E59   . 83E8 00         SUB EAX,0                              Switch (cases 0..1)
00457E5C   .v74 1B           JE SHORT Gw.00457E79
00457E5E   . 48              DEC EAX
00457E5F   .v74 14           JE SHORT Gw.00457E75
00457E61   . 68 721E0000     PUSH 1E72                              Arg1 = 00001E72; Defau
00457E66   . BA 743B9D00     MOV EDX,Gw.009D3B74                    ASCII "GmView.cpp"
00457E6B   . B9 A40CA300     MOV ECX,Gw.00A30CA4
00457E70   . E8 5BA8FAFF     CALL Gw.004026D0                       Gw.004026D0
00457E75   > 33C9            XOR ECX,ECX                            Case 1 of switch 00457
00457E77   .vEB 05           JMP SHORT Gw.00457E7E
00457E79   > B9 01000000     MOV ECX,1                              Case 0 of switch 00457
00457E7E   > 56              PUSH ESI
00457E7F   . E8 3C000000     CALL Gw.00457EC0
00457E84   . 8BF0            MOV ESI,EAX
00457E86   . 8BCF            MOV ECX,EDI
00457E88   . 8D56 11         LEA EDX,DWORD PTR DS:[ESI+11]
00457E8B   . E8 10C61A00     CALL Gw.006044A0
00457E90   . 85C0            TEST EAX,EAX
00457E92   .v74 0D           JE SHORT Gw.00457EA1
00457E94   . 6A 00           PUSH 0                                 Arg2 = 00000000
```

```
Registers (FPU)
EAX 1916DD20
ECX 00000007
EDX 03A3FA88
EBX 03A3FA88
ESP 03A3F77C
EBP 03A3F798
ESI 0000000C
EDI 00000007

EIP 00457E56 Gw.00457E56

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 2D0000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_NOT_ENOUGH_MEMORY

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

ST0 empty 1.0000000000000000000
ST1 empty 1.0000000000000000000
ST2 empty 0.0
ST3 empty -24887.902349370997400
ST4 empty -12525.209502177134710
```

```
Stack DS:[03A3FA8C]=00000000
EAX=1916DD20
```

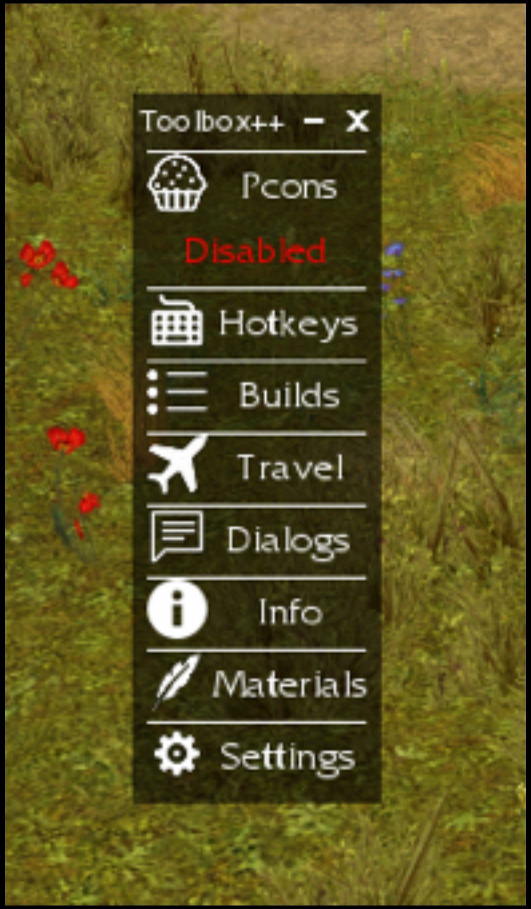
```
Address   Hex dump                  ASCII
1676C380  54 00 65 00 6D 00 70 00   T.e.m.p.
1676C388  6C 00 61 00 74 00 65 00   l.a.t.e.
1676C390  20 00 74 00 65 00 73 00    .t.e.s.
1676C398  74 00 21 00 00 00 00 00   t.!.....
1676C3A0  79 F4 16 6E D3 7A 00 88   yô▯nÓz.ˆ
1676C3A8  3C C0 6E 0C 3A 00 00 00   <Àn.:...
1676C3B0  40 BB 76 16 D8 C0 76 16   @»v▯ØÀv▯
1676C3B8  58 C0 6E 0C 49 C0 6E 0C   XÀn.IÀn.
1676C3C0  00 00 40 41 1F 60 AC 3B   ..@A`¬;
```

```
03A3FA88  00000000
03A3FA8C  00000000
03A3FA90  0CF65DB8  UNICODE "OwFj0xfzITOMMMHMxgxZ6P0k40A"
03A3FA94  0000001C
03A3FA98  0000001C
03A3FA9C  00000080
03A3FAA0  1676C380  UNICODE "Template test!"
03A3FAA4  03A3FABC
03A3FAA8  004A6847  RETURN to Gw.004A6847 from Gw.004A4F90
03A3FAAC  0D5B2510
```

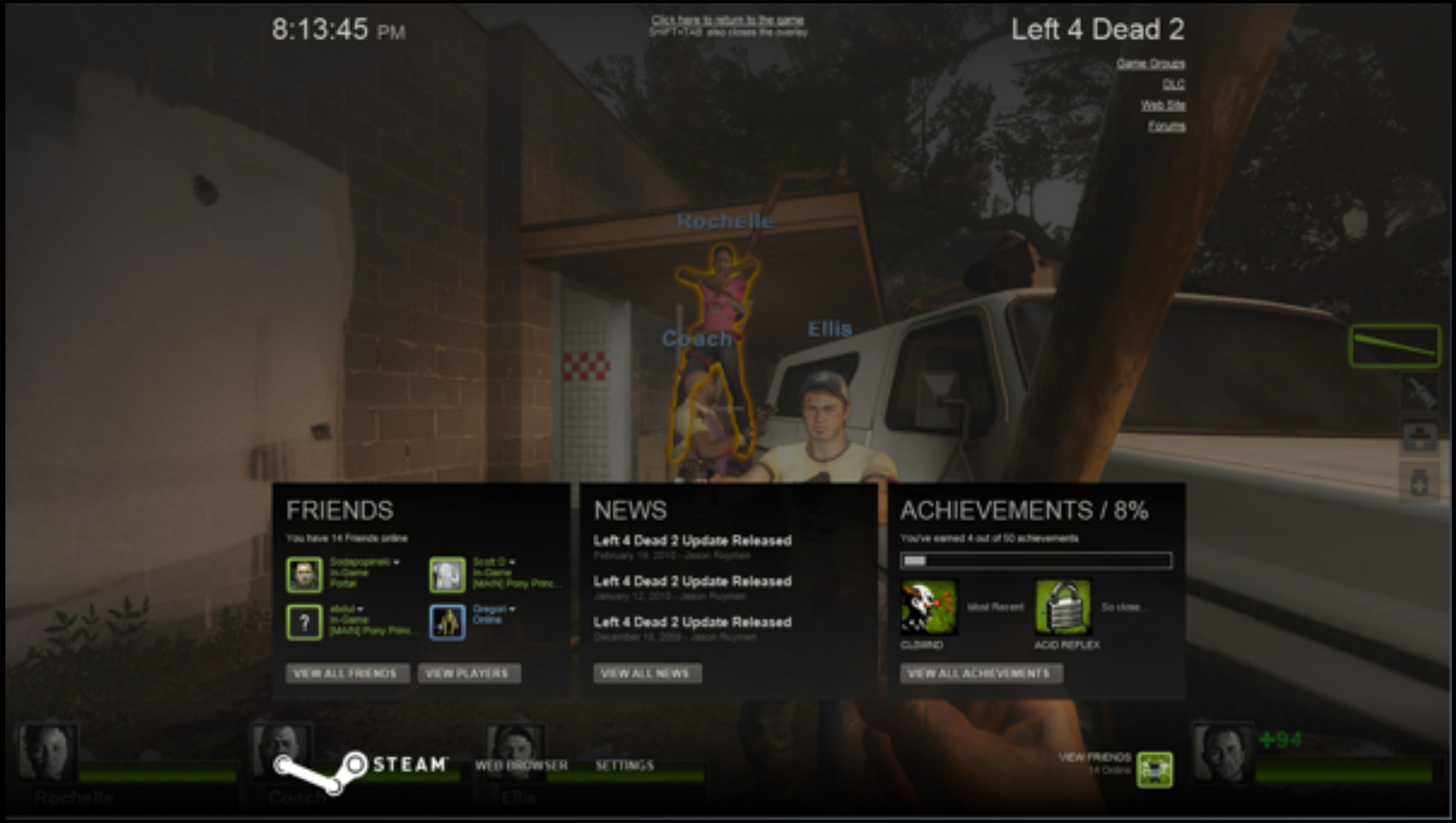
```
Breakpoint at Gw.00457E56                                      Paused
```

DirectX EndScene Hooking

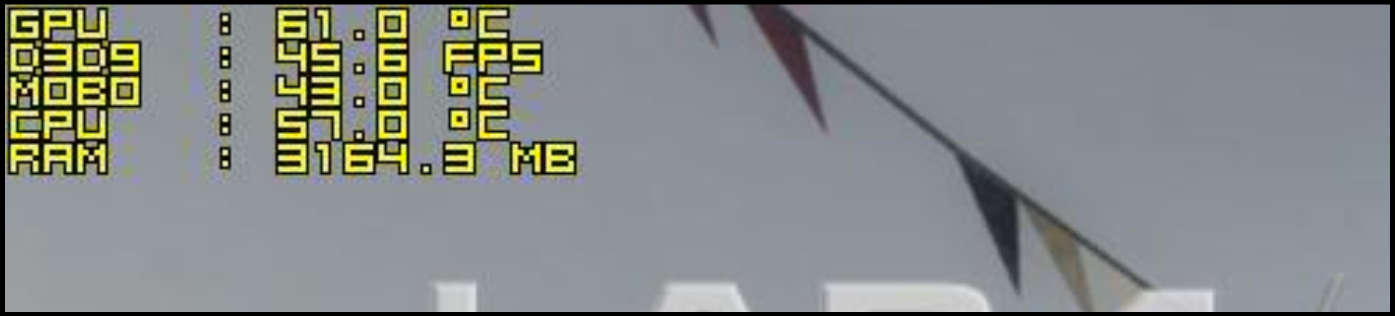
Game Mods



Steam Overlay


8:13:45 PM
Left 4 Dead 2
FRIENDS
NEWS
Left 4 Dead 2 Update Released
Left 4 Dead 2 Update Released
Left 4 Dead 2 Update Released
ACHIEVEMENTS / 8%
STEAM
WEB BROWSER
SETTINGS

Performance Monitors



```
GPU     :  61.0  °C
D3D9    :  45.6  FPS
MOBO    :  43.0  °C
CPU     :  57.0  °C
RAM     :  3164.3  MB
```

FPS Counters

DLL injection and x86 hooking demystified

Other topics include:

- Advanced / Stealth injection techniques
- Integrity of execution during hook installation
- Hook restoration / cleanup
- Hooking detection (anti-cheat) and advanced hooking methods
- Multiple layers of hooks
- Prevent hook recursion
- Hooking different calling conventions and class methods

Sources:

What is a DLL?
https://support.microsoft.com/en-ca/kb/815065

Windows DLL Injection Basics by Brad Antoniewicz
http://blog.opensecurityresearch.com/2013/01/windows-dll-injection-basics.html

x86 API Hooking Demystified by Jurriaan Bremer
http://jbremer.org/x86-api-hooking-demystified/