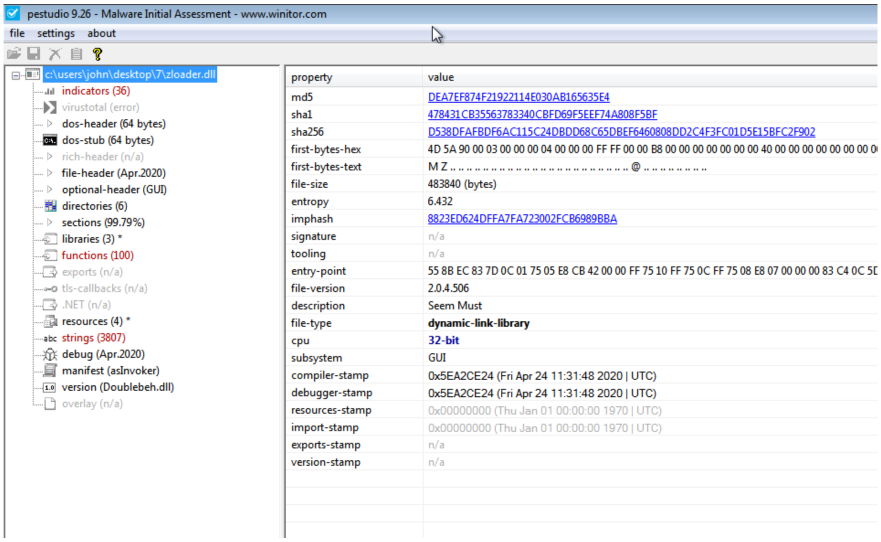
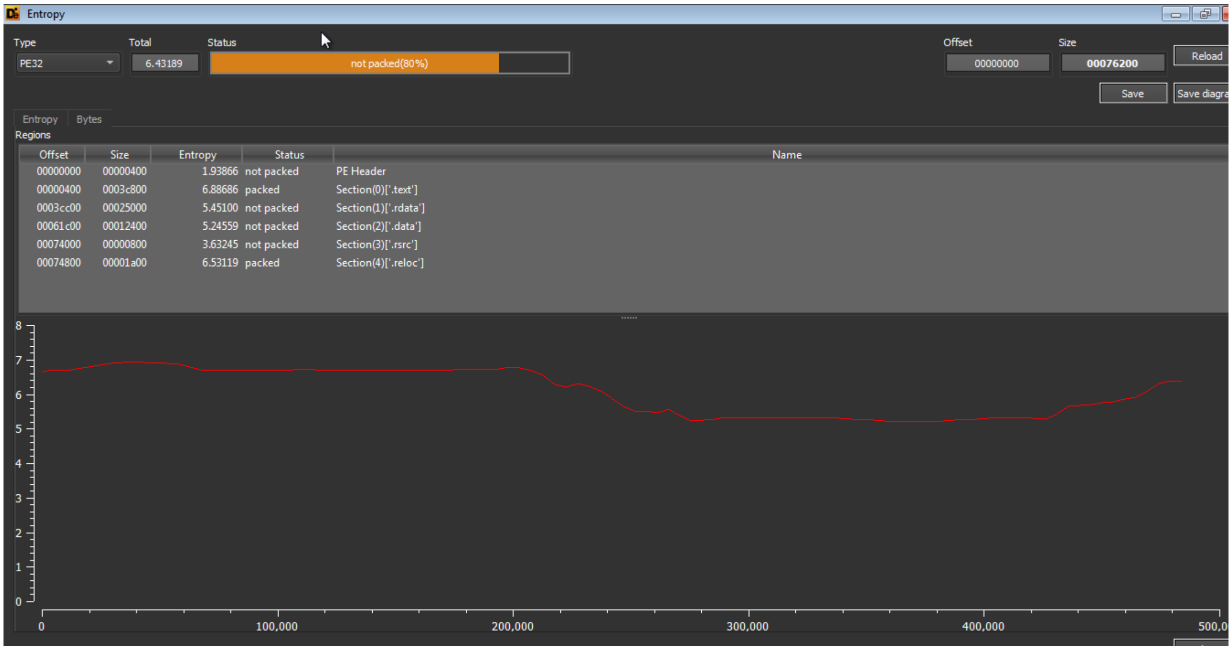
Initially we loaded the sample in pestudio.Here entropy is less so we can assume that sample is unpacked.



property	value
md5	DEA7EF874F21922114E030AB165635E4
sha1	478431CB35563783340CBFD69F5EEF74A808F5BF
sha256	D538DFAFBDF6AC115C24DBDD68C65DBEF6460808DD2C4F3FC01D5E15BFC2F902
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 0
first-bytes-text	M Z .. .. @ .. ..
file-size	483840 (bytes)
entropy	6.432
imphash	8823ED624DFFA7FA723002FCB6989BBA
signature	n/a
tooling	n/a
entry-point	55 8B EC 83 7D 0C 01 75 05 E8 CB 42 00 00 FF 75 10 FF 75 0C FF 75 08 E8 07 00 00 00 83 C4 0C 5D
file-version	2.0.4.506
description	Seem Must
file-type	**dynamic-link-library**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x5EA2CE24 (Fri Apr 24 11:31:48 2020 | UTC)
debugger-stamp	0x5EA2CE24 (Fri Apr 24 11:31:48 2020 | UTC)
resources-stamp	0x00000000 (Thu Jan 01 00:00:00 1970 | UTC)
import-stamp	0x00000000 (Thu Jan 01 00:00:00 1970 | UTC)
exports-stamp	n/a
version-stamp	n/a

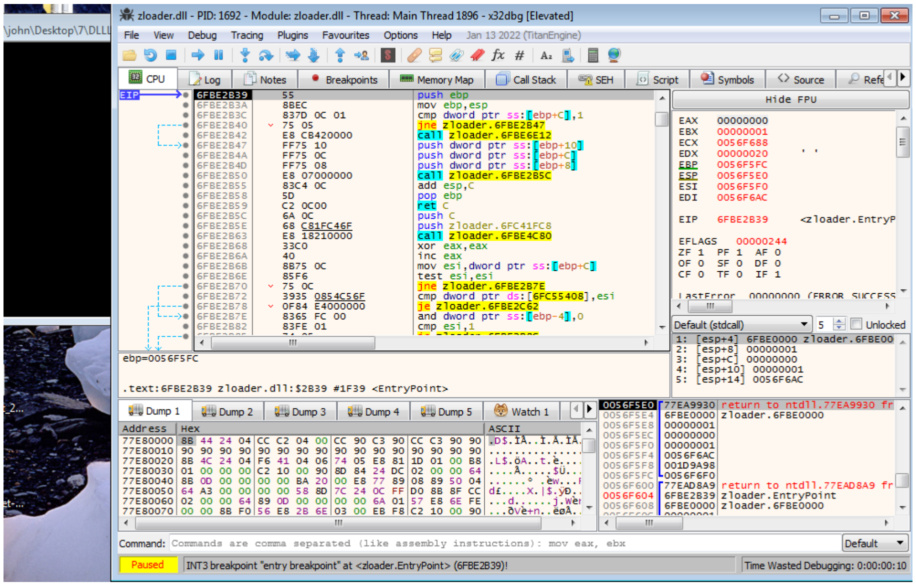
Also in Detect It Easy it is showing unpacked.



Type: PE32
Total: 6.43189
Status: not packed(80%)
Offset: 00000000
Size: **00076200**

Offset	Size	Entropy	Status	Name
00000000	00000400	1.93866	not packed	PE Header
00000400	0003c800	6.88686	packed	Section(0)['.text']
0003cc00	00025000	5.45100	not packed	Section(1)['.rdata']
00061c00	00012400	5.24559	not packed	Section(2)['.data']
00074000	00000800	3.63245	not packed	Section(3)['.rsrc']
00074800	00001a00	6.53119	packed	Section(4)['.reloc']

Here we uploaded the sample in x32dbg as we load the sample immediately the dll loader is started in order to load dlls.


zloader.dll - PID: 1692 - Module: zloader.dll - Thread: Main Thread 1896 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source Refe
Hide FPU
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1
Default (stdcall) 5 Unlocked
Default

```
EIP  6FBE2B39   55                push ebp
     6FBE2B3A   8BEC              mov ebp,esp
     6FBE2B3C   837D 0C 01        cmp dword ptr ss:[ebp+C],1
     6FBE2B40   75 05             jne zloader.6FBE2B47
     6FBE2B42   E8 CB420000       call zloader.6FBE6E12
     6FBE2B47   FF75 10           push dword ptr ss:[ebp+10]
     6FBE2B4A   FF75 0C           push dword ptr ss:[ebp+C]
     6FBE2B4D   FF75 08           push dword ptr ss:[ebp+8]
     6FBE2B50   E8 07000000       call zloader.6FBE2B5C
     6FBE2B55   83C4 0C           add esp,C
     6FBE2B58   5D                pop ebp
     6FBE2B59   C2 0C00           ret C
     6FBE2B5C   6A 0C             push C
     6FBE2B5E   68 C81FC46F       push zloader.6FC41FC8
     6FBE2B63   E8 18210000       call zloader.6FBE4C80
     6FBE2B68   33C0              xor eax,eax
     6FBE2B6A   40                inc eax
     6FBE2B6B   8B75 0C           mov esi,dword ptr ss:[ebp+C]
     6FBE2B6E   85F6              test esi,esi
     6FBE2B70   75 0C             jne zloader.6FBE2B7E
     6FBE2B72   3935 0854C56F     cmp dword ptr ds:[6FC55408],esi
     6FBE2B78   0F84 E4000000     je zloader.6FBE2C62
     6FBE2B7E   8365 FC 00        and dword ptr ss:[ebp-4],0
     6FBE2B82   83FE 01           cmp esi,1

ebp=0056F5FC

.text:6FBE2B39 zloader.dll:$2B39 #1F39 <EntryPoint>
```

```
EAX   00000000
EBX   00000001
ECX   0056F688
EDX   00000020     ' '
EBP   0056F5FC
ESP   0056F5E0
ESI   0056F5F0
EDI   0056F6AC

EIP   6FBE2B39     <zloader.EntryP

EFLAGS   00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS

1: [esp+4]  6FBE0000 zloader.6FBE00
2: [esp+8]  00000001
3: [esp+C]  00000000
4: [esp+10] 00000001
5: [esp+14] 0056F6AC
```

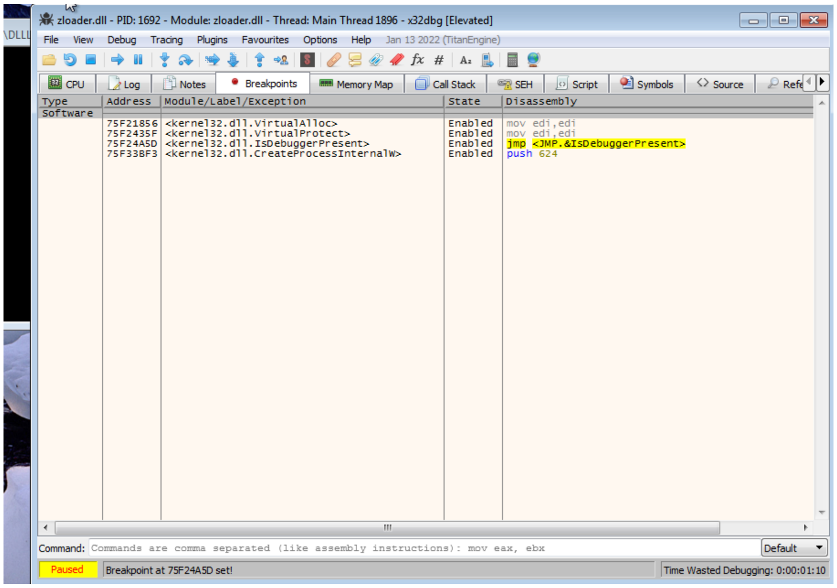
```
Address   Hex                                              ASCII
77E80000  8B 44 24 04 CC C2 04 00 CC 90 C3 90 CC C3 90 90  .D$.ÌÂ..Ì.Ã.ÌÃ.
77E80010  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
77E80020  8B 4C 24 04 F6 41 04 06 74 05 E8 81 1D 01 00 B8  .L$.öA..t.è.....
77E80030  01 00 00 00 C2 10 00 90 8D 84 24 DC 02 00 00 64  ....Â.....$Ü...d
77E80040  8B 0D 00 00 00 00 BA 20 00 E8 77 89 08 89 50 04  ......º .èw...P.
77E80050  64 A3 00 00 00 00 58 8D 7C 24 0C FF D0 8B 8F CC  d£....X.|$.ÿÐ..Ì
77E80060  02 00 00 64 89 0D 00 00 00 00 6A 01 57 E8 6E FE  ...d......j.Wèn.
77E80070  00 00 8B F0 56 E8 2B 6E 03 00 EB F8 C2 10 00 90  ..ðVè+n..ëøÂ...
```

```
0056F5E0  77EA9930  return to ntdll.77EA9930 fr
0056F5E4  6FBE0000  zloader.6FBE0000
0056F5E8  00000001
0056F5EC  00000000
0056F5F0  00000001
0056F5F4  0056F6AC
0056F5F8  001D9A98
0056F5FC  0056F6F0
0056F600  77EAD8A9  return to ntdll.77EAD8A9 fr
0056F604  6FBE2B39  zloader.EntryPoint
0056F608  6FBE0000  zloader.6FBE0000
```

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

Paused INT3 breakpoint "entry breakpoint" at <zloader.EntryPoint> (6FBE2B39)! Time Wasted Debugging: 0:00:00:10

We put breakpoints on VirtualAlloc, VirtualProtect, IsDebuggerPresent, CreateProcessInternalW, now press f9.

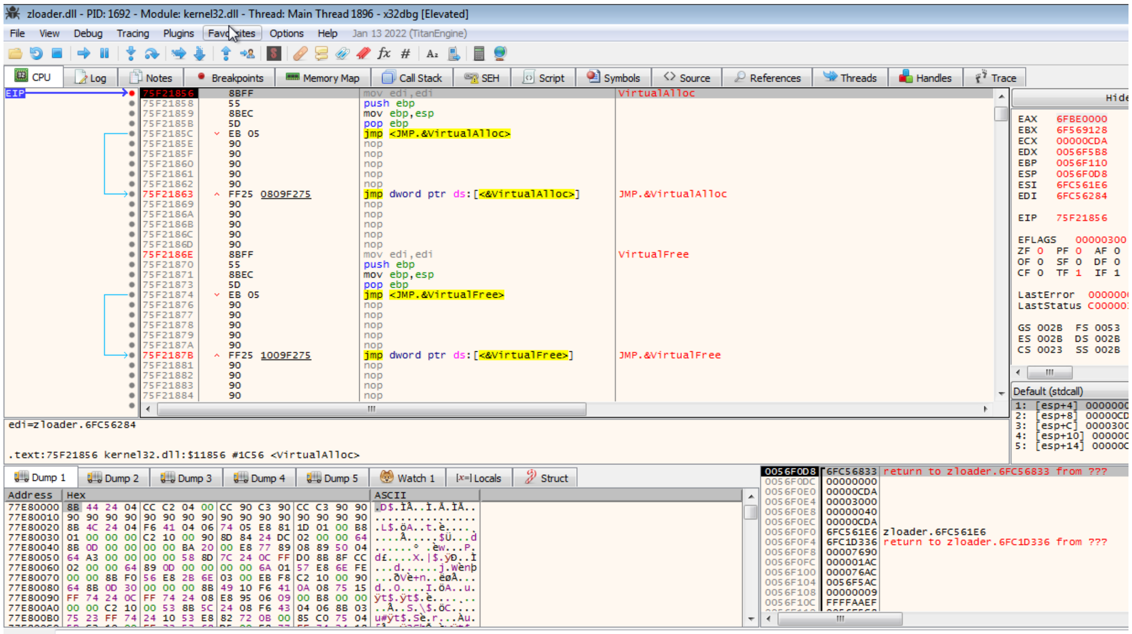

zloader.dll - PID: 1692 - Module: zloader.dll - Thread: Main Thread 1896 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source Refe
Default

Type	Address	Module/Label/Exception	State	Disassembly
Software				
	75F21856	<kernel32.dll.VirtualAlloc>	Enabled	mov edi,edi
	75F2435F	<kernel32.dll.VirtualProtect>	Enabled	mov edi,edi
	75F24A5D	<kernel32.dll.IsDebuggerPresent>	Enabled	jmp <JMP.&IsDebuggerPresent>
	75F33BF3	<kernel32.dll.CreateProcessInternalW>	Enabled	push 624

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

Paused Breakpoint at 75F24A5D set! Time Wasted Debugging: 0:00:01:10

Here we hit our first breakpoint VirtualAlloc, just step over it.



zloader.dll - PID: 1692 - Module: kernel32.dll - Thread: Main Thread 1896 - x32dbg [Elevated]

```
EIP  75F21856    8BFF           mov edi,edi                          VirtualAlloc
     75F21858    55             push ebp
     75F21859    8BEC           mov ebp,esp
     75F2185B    5D             pop ebp
     75F2185C    EB 05          jmp <JMP.&VirtualAlloc>
     75F2185E    90             nop
     75F2185F    90             nop
     75F21860    90             nop
     75F21861    90             nop
     75F21862    90             nop
     75F21863    FF25 0809F275  jmp dword ptr ds:[<&VirtualAlloc>]   JMP.&VirtualAlloc
     75F21869    90             nop
     75F2186A    90             nop
     75F2186B    90             nop
     75F2186C    90             nop
     75F2186D    90             nop
     75F2186E    8BFF           mov edi,edi                          VirtualFree
     75F21870    55             push ebp
     75F21871    8BEC           mov ebp,esp
     75F21873    5D             pop ebp
     75F21874    EB 05          jmp <JMP.&VirtualFree>
     75F21876    90             nop
     75F21877    90             nop
     75F21878    90             nop
     75F21879    90             nop
     75F2187A    90             nop
     75F2187B    FF25 1009F275  jmp dword ptr ds:[<&VirtualFree>]    JMP.&VirtualFree
     75F21881    90             nop
     75F21882    90             nop
     75F21883    90             nop
     75F21884    90             nop
```

```
edi=zloader.6FC56284

.text:75F21856 kernel32.dll:$11856 #1C56 <VirtualAlloc>
```

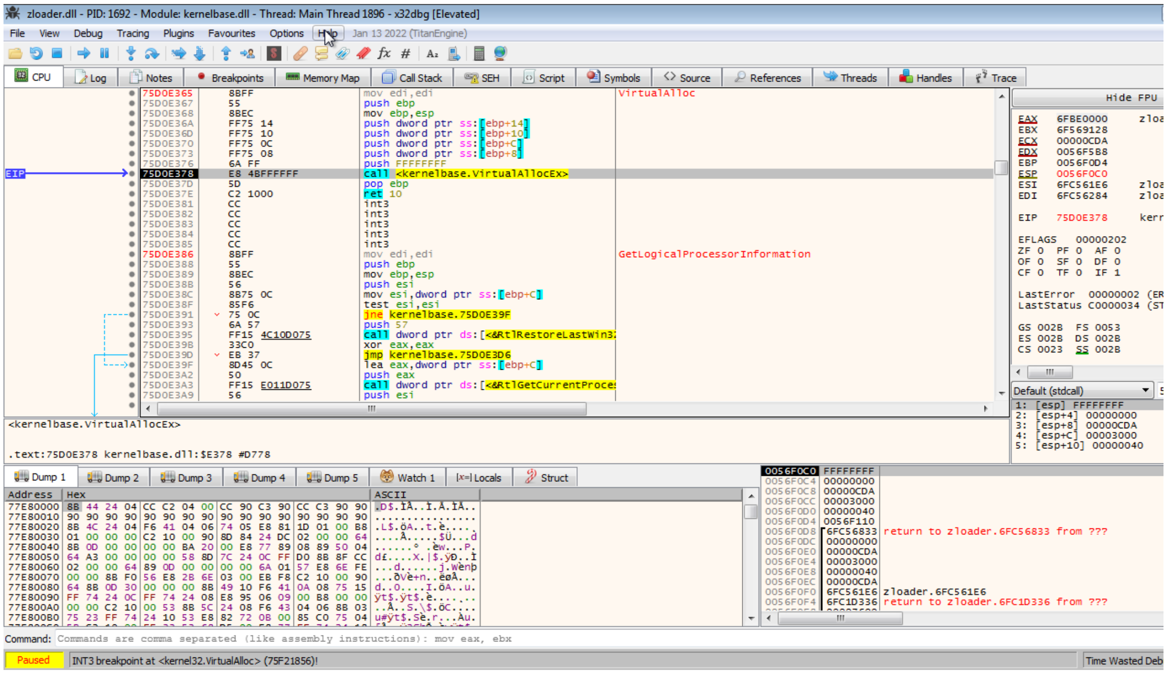
```
EAX    6FBE0000
EBX    6F569128
ECX    00000CDA
EDX    0056F5B8
EBP    0056F110
ESP    0056F0D8
ESI    6FC561E6
EDI    6FC56284

EIP    75F21856

EFLAGS    00000300
ZF 0   PF 0   AF 0
OF 0   SF 0   DF 0
CF 0   TF 1   IF 1

LastError   000000
LastStatus  C00000

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

```
1: [esp+4]  000000
2: [esp+8]  00000CD
3: [esp+C]  000030C
4: [esp+10] 00000C
5: [esp+14] 00000C
```

```
Address   Hex                                              ASCII
77E80000  8B 44 24 04 CC C2 04 00 CC 90 C3 90 CC C3 90 90  .D$.ÌÂ..Ì.Ã.ÌÃ..
77E80010  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
77E80020  8B 4C 24 04 F6 41 04 06 74 05 E8 81 1D 01 00 B8  .L$.öA..t.è.....
77E80030  01 00 00 00 C2 10 00 90 8D 84 24 DC 02 00 00 64  ....Â.....$Ü...d
77E80040  8B 0D 00 00 00 00 BA 20 00 E8 77 89 08 89 50 04  ......º .èw...P.
77E80050  64 A3 00 00 00 00 58 8D 7C 24 0C FF D0 8B 8F CC  d£....X.|$.ÿÐ..Ì
77E80060  02 00 00 64 89 0D 00 00 00 00 6A 01 57 E8 6E FE  ...d......j.Wènþ
77E80070  00 00 8B F0 56 E8 2B 6E 03 00 EB F8 C2 10 00 90  ...ðVè+n..ëøÂ...
77E80080  64 8B 0D 30 00 00 00 8B 49 10 F6 41 0A 08 75 15  d..0....I.öA..u.
77E80090  FF 74 24 0C FF 74 24 08 E8 95 06 09 00 B8 00 00  ÿt$.ÿt$.è.......
77E800A0  00 00 C2 10 00 53 8B 5C 24 08 F6 43 04 06 8B 03  ..Â..S.\$.öC....
77E800B0  75 23 FF 74 24 10 53 E8 82 72 0B 00 85 C0 75 04  u#ÿt$.Sè.r...Àu.
```

```
0056F0D8  6FC56833  return to zloader.6FC56833 from ???
0056F0DC  00000000
0056F0E0  00000CDA
0056F0E4  00003000
0056F0E8  00000040
0056F0EC  00000CDA
0056F0F0  6FC561E6  zloader.6FC561E6
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8  00007690
0056F0FC  000001AC
0056F100  000076AC
0056F104  0056F5AC
0056F108  00000009
0056F10C  FFFFAAEF
```

Now here just step over it and dump the address return by it in eax.



zloader.dll - PID: 1692 - Module: kernelbase.dll - Thread: Main Thread 1896 - x32dbg [Elevated]

```
     75D0E365    8BFF           mov edi,edi                          VirtualAlloc
     75D0E367    55             push ebp
     75D0E368    8BEC           mov ebp,esp
     75D0E36A    FF75 14        push dword ptr ss:[ebp+14]
     75D0E36D    FF75 10        push dword ptr ss:[ebp+10]
     75D0E370    FF75 0C        push dword ptr ss:[ebp+C]
     75D0E373    FF75 08        push dword ptr ss:[ebp+8]
     75D0E376    6A FF          push FFFFFFFF
EIP  75D0E378    E8 4BFFFFFF    call <kernelbase.VirtualAllocEx>
     75D0E37D    5D             pop ebp
     75D0E37E    C2 1000        ret 10
     75D0E381    CC             int3
     75D0E382    CC             int3
     75D0E383    CC             int3
     75D0E384    CC             int3
     75D0E385    CC             int3
     75D0E386    8BFF           mov edi,edi                          GetLogicalProcessorInformation
     75D0E388    55             push ebp
     75D0E389    8BEC           mov ebp,esp
     75D0E38B    56             push esi
     75D0E38C    8B75 0C        mov esi,dword ptr ss:[ebp+C]
     75D0E38F    85F6           test esi,esi
     75D0E391    75 0C          jne kernelbase.75D0E39F
     75D0E393    6A 57          push 57
     75D0E395    FF15 4C10D075  call dword ptr ds:[<&RtlRestoreLastWin32
     75D0E39B    33C0           xor eax,eax
     75D0E39D    EB 37          jmp kernelbase.75D0E3D6
     75D0E39F    8D45 0C        lea eax,dword ptr ss:[ebp+C]
     75D0E3A2    50             push eax
     75D0E3A3    FF15 E011D075  call dword ptr ds:[<&RtlGetCurrentProces
     75D0E3A9    56             push esi
```

```
<kernelbase.VirtualAllocEx>

.text:75D0E378 kernelbase.dll:$E378 #D778
```

```
EAX    6FBE0000     zloa
EBX    6F569128
ECX    00000CDA
EDX    0056F5B8
EBP    0056F0D4
ESP    0056F0C0
ESI    6FC561E6     zloa
EDI    6FC56284     zloa

EIP    75D0E378     kerr

EFLAGS    00000202
ZF 0   PF 0   AF 0
OF 0   SF 0   DF 0
CF 0   TF 0   IF 1

LastError   00000002 (ER
LastStatus  C0000034 (ST

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

```
1: [esp]    FFFFFFFF
2: [esp+4]  00000000
3: [esp+8]  00000CDA
4: [esp+C]  00003000
5: [esp+10] 00000040
```

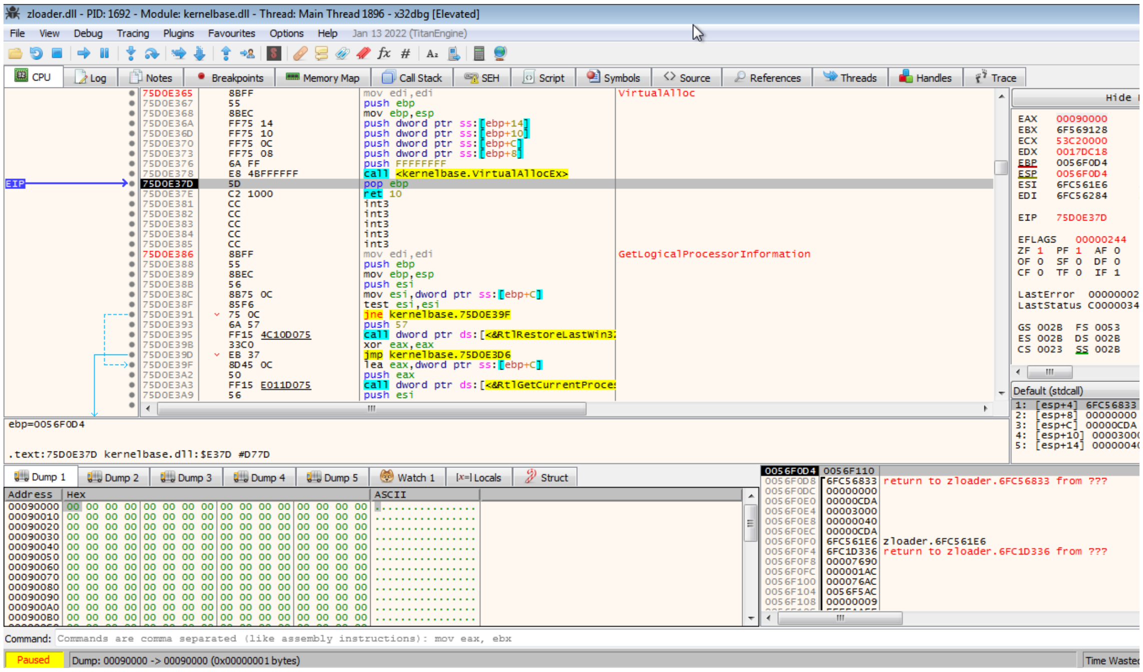
```
Address   Hex                                              ASCII
77E80000  8B 44 24 04 CC C2 04 00 CC 90 C3 90 CC C3 90 90  .D$.ÌÂ..Ì.Ã.ÌÃ..
77E80010  90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
77E80020  8B 4C 24 04 F6 41 04 06 74 05 E8 81 1D 01 00 B8  .L$.öA..t.è.....
77E80030  01 00 00 00 C2 10 00 90 8D 84 24 DC 02 00 00 64  ....Â.....$Ü...d
77E80040  8B 0D 00 00 00 00 BA 20 00 E8 77 89 08 89 50 04  ......º .èw...P.
77E80050  64 A3 00 00 00 00 58 8D 7C 24 0C FF D0 8B 8F CC  d£....X.|$.ÿÐ..Ì
77E80060  02 00 00 64 89 0D 00 00 00 00 6A 01 57 E8 6E FE  ...d......j.Wènþ
77E80070  00 00 8B F0 56 E8 2B 6E 03 00 EB F8 C2 10 00 90  ...ðVè+n..ëøÂ...
77E80080  64 8B 0D 30 00 00 00 8B 49 10 F6 41 0A 08 75 15  d..0....I.öA..u.
77E80090  FF 74 24 0C FF 74 24 08 E8 95 06 09 00 B8 00 00  ÿt$.ÿt$.è.......
77E800A0  00 00 C2 10 00 53 8B 5C 24 08 F6 43 04 06 8B 03  ..Â..S.\$.öC....
77E800B0  75 23 FF 74 24 10 53 E8 82 72 0B 00 85 C0 75 04  u#ÿt$.Sè.r...Àu.
```

```
0056F0C0  FFFFFFFF
0056F0C4  00000000
0056F0C8  00000CDA
0056F0CC  00003000
0056F0D0  00000040
0056F0D4  0056F110
0056F0D8  6FC56833  return to zloader.6FC56833 from ???
0056F0DC  00000000
0056F0E0  00000CDA
0056F0E4  00003000
0056F0E8  00000040
0056F0EC  00000CDA
0056F0F0  6FC561E6  zloader.6FC561E6
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
```

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

Paused   INT3 breakpoint at <kernel32.VirtualAlloc> (75F21856)!

Here the address return is 90000000 in eax and we dump that in dump1 and then press f9.



zloader.dll - PID: 1692 - Module: kernelbase.dll - Thread: Main Thread 1896 - x32dbg [Elevated]

```
75D0E365  8BFF            mov edi,edi                                   VirtualAlloc
75D0E367  55              push ebp
75D0E368  8BEC            mov ebp,esp
75D0E36A  FF75 14         push dword ptr ss:[ebp+14]
75D0E36D  FF75 10         push dword ptr ss:[ebp+10]
75D0E370  FF75 0C         push dword ptr ss:[ebp+C]
75D0E373  FF75 08         push dword ptr ss:[ebp+8]
75D0E376  6A FF           push FFFFFFFF
75D0E378  E8 4BFFFFFF     call <kernelbase.VirtualAllocEx>
75D0E37D  5D              pop ebp
75D0E37E  C2 1000         ret 10
75D0E381  CC              int3
75D0E382  CC              int3
75D0E383  CC              int3
75D0E384  CC              int3
75D0E385  CC              int3
75D0E386  8BFF            mov edi,edi                                   GetLogicalProcessorInformation
75D0E388  55              push ebp
75D0E389  8BEC            mov ebp,esp
75D0E38B  56              push esi
75D0E38C  8B75 0C         mov esi,dword ptr ss:[ebp+C]
75D0E38F  85F6            test esi,esi
75D0E391  75 0C           jne kernelbase.75D0E39F
75D0E393  6A 57           push 57
75D0E395  FF15 4C10D075   call dword ptr ds:[<&RtlRestoreLastWin3
75D0E39B  33C0            xor eax,eax
75D0E39D  EB 37           jmp kernelbase.75D0E3D6
75D0E39F  8D45 0C         lea eax,dword ptr ss:[ebp+C]
75D0E3A2  50              push eax
75D0E3A3  FF15 E011D075   call dword ptr ds:[<&RtlGetCurrentProce
75D0E3A9  56              push esi

ebp=0056F0D4

.text:75D0E37D kernelbase.dll:$E37D #D77D
```

```
EAX  00090000
EBX  6F569128
ECX  53C20000
EDX  0017DC18
EBP  0056F0D4
ESP  0056F0D4
ESI  6FC561E6
EDI  6FC56284

EIP  75D0E37D

EFLAGS  00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000002
LastStatus C0000034

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

Default (stdcall)
1: [esp+4] 6FC56833
2: [esp+8] 00000000
3: [esp+C] 00000CDA
4: [esp+10] 0000300
5: [esp+14] 0000004
```

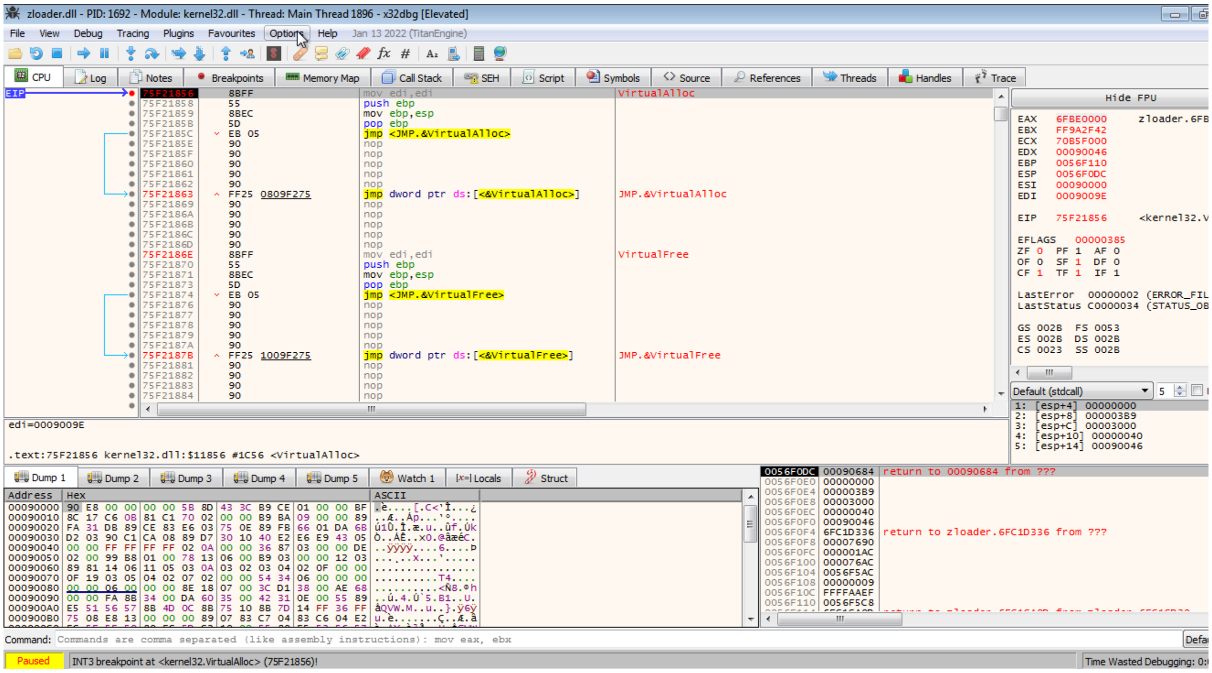
```
Address   Hex                                              ASCII
00090000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00090010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00090020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00090030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00090040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00090050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00090060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00090070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00090080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00090090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000900A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000900B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
0056F0D4  0056F110
0056F0D8  6FC56833  return to zloader.6FC56833 from ???
0056F0DC  00000000
0056F0E0  00000CDA
0056F0E4  00003000
0056F0E8  00000040
0056F0EC  00000CDA
0056F0F0  6FC561E6  zloader.6FC561E6
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8  00007690
0056F0FC  000001AC
0056F100  000076AC
0056F104  0056F5AC
0056F108  00000009
```

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

Paused Dump: 00090000 -> 00090000 (0x00000001 bytes)

Here we again hit VirtualAlloc now dump1 is filled with values here just step over .



zloader.dll - PID: 1692 - Module: kernel32.dll - Thread: Main Thread 1896 - x32dbg [Elevated]

```
75F21856  8BFF            mov edi,edi                                   VirtualAlloc
75F21858  55              push ebp
75F21859  8BEC            mov ebp,esp
75F2185B  5D              pop ebp
75F2185C  EB 05           jmp <JMP.&VirtualAlloc>
75F2185E  90              nop
75F2185F  90              nop
75F21860  90              nop
75F21861  90              nop
75F21862  90              nop
75F21863  FF25 0809F275   jmp dword ptr ds:[<&VirtualAlloc>]           JMP.&VirtualAlloc
75F21869  90              nop
75F2186A  90              nop
75F2186B  90              nop
75F2186C  90              nop
75F2186D  90              nop
75F2186E  8BFF            mov edi,edi                                   VirtualFree
75F21870  55              push ebp
75F21871  8BEC            mov ebp,esp
75F21873  5D              pop ebp
75F21874  EB 05           jmp <JMP.&VirtualFree>
75F21876  90              nop
75F21877  90              nop
75F21878  90              nop
75F21879  90              nop
75F2187A  90              nop
75F2187B  FF25 1009F275   jmp dword ptr ds:[<&VirtualFree>]            JMP.&VirtualFree
75F21881  90              nop
75F21882  90              nop
75F21883  90              nop
75F21884  90              nop

edi=0009009E

.text:75F21856 kernel32.dll:$11856 #1C56 <VirtualAlloc>
```

```
EAX  6FBE0000     zloader.6FB
EBX  FF9A2F42
ECX  70B5F000
EDX  00090046
EBP  0056F110
ESP  0056F0DC
ESI  00090000
EDI  0009009E

EIP  75F21856     <kernel32.V

EFLAGS  00000385
ZF 0  PF 1  AF 0
OF 0  SF 1  DF 0
CF 1  TF 1  IF 1

LastError  00000002 (ERROR_FIL
LastStatus C0000034 (STATUS_OB

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

Default (stdcall)  5
1: [esp+4] 00000000
2: [esp+8] 000003B9
3: [esp+C] 00003000
4: [esp+10] 00000040
5: [esp+14] 00090046
```

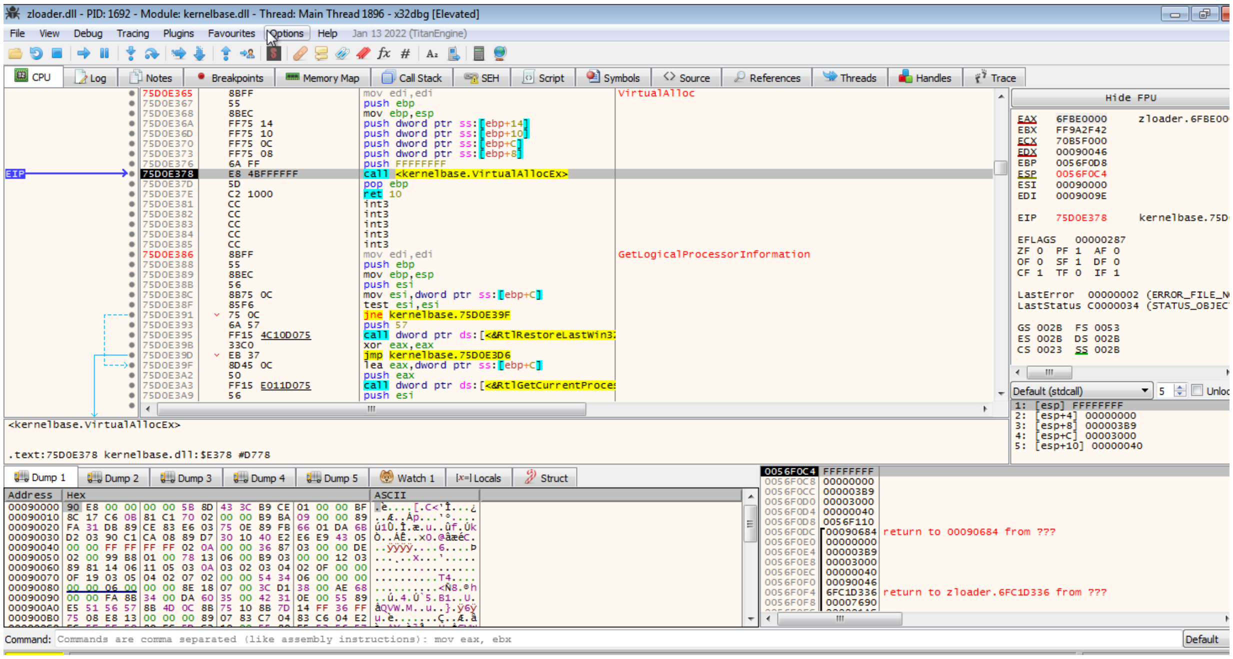
```
Address   Hex                                              ASCII
00090000  90 E8 00 00 00 00 5B 8D 43 3C B9 CE 01 00 00 BF  .è....[.C<'Î...¿
00090010  8C 17 C6 0B 81 C1 70 02 00 00 B9 BA 09 00 00 89  ..Æ..Áp...'°....
00090020  FA 31 DB 89 CE 83 E6 03 75 0E 89 FB 66 01 DA 6B  ú1Û.Î.æ.u..ûf.Úk
00090030  D2 03 90 C1 CA 08 89 D7 30 10 40 E2 E6 E9 43 05  Ò..ÁÊ..×0.@âæéC.
00090040  00 00 FF FF FF FF 02 0A 00 00 36 87 03 00 00 DE  ..ÿÿÿÿ....6....Þ
00090050  02 00 99 B8 01 00 78 13 06 00 B9 03 00 00 12 03  ......x...'.....
00090060  89 81 14 06 11 05 03 0A 03 02 03 04 02 0F 00 00  ................
00090070  0F 19 03 05 04 02 07 02 00 00 54 34 06 00 00 00  ..........T4....
00090080  00 00 06 00 00 00 8E 18 07 00 3C D1 38 00 AE 68  ..........<Ñ8.®h
00090090  00 00 FA 8B 34 00 DA 60 35 00 42 31 0E 00 55 89  ..ú.4.Ú`5.B1..U.
000900A0  E5 51 56 57 8B 4D 0C 8B 75 10 8B 7D 14 FF 36 FF  åQVW.M..u..}.ÿ6ÿ
000900B0  75 08 E8 13 00 00 00 89 07 83 C7 04 83 C6 04 E2  u.è.......Ç..Æ.â
```

```
0056F0DC  00090684  return to 00090684 from ???
0056F0E0  00000000
0056F0E4  000003B9
0056F0E8  00003000
0056F0EC  00000040
0056F0F0  00090046
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8  00007690
0056F0FC  000001AC
0056F100  000076AC
0056F104  0056F5AC
0056F108  00000009
0056F10C  FFFFAAEF
0056F110  0056F5C8
```

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

Paused INT3 breakpoint at <kernel32.VirtualAlloc> (75F21856)! Time Wasted Debugging: 0:

After this call the address will be return in eax , dump that memory region in dump2.


zloader.dll - PID: 1692 - Module: kernelbase.dll - Thread: Main Thread 1896 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
Hide FPU
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct
Default (stdcall) 5 Unloc
Default

```
     75D0E365   8BFF           mov edi,edi                              VirtualAlloc
     75D0E367   55             push ebp
     75D0E368   8BEC           mov ebp,esp
     75D0E36A   FF75 14        push dword ptr ss:[ebp+14]
     75D0E36D   FF75 10        push dword ptr ss:[ebp+10]
     75D0E370   FF75 0C        push dword ptr ss:[ebp+C]
     75D0E373   FF75 08        push dword ptr ss:[ebp+8]
     75D0E376   6A FF          push FFFFFFFF
EIP  75D0E378   E8 4BFFFFFF    call <kernelbase.VirtualAllocEx>
     75D0E37D   5D             pop ebp
     75D0E37E   C2 1000        ret 10
     75D0E381   CC             int3
     75D0E382   CC             int3
     75D0E383   CC             int3
     75D0E384   CC             int3
     75D0E385   CC             int3
     75D0E386   8BFF           mov edi,edi                              GetLogicalProcessorInformation
     75D0E388   55             push ebp
     75D0E389   8BEC           mov ebp,esp
     75D0E38B   56             push esi
     75D0E38C   8B75 0C        mov esi,dword ptr ss:[ebp+C]
     75D0E38F   85F6           test esi,esi
     75D0E391   75 0C          jne kernelbase.75D0E39F
     75D0E393   6A 57          push 57
     75D0E395   FF15 4C10D075  call dword ptr ds:[<&RtlRestoreLastWin3
     75D0E39B   33C0           xor eax,eax
     75D0E39D   EB 37          jmp kernelbase.75D0E3D6
     75D0E39F   8D45 0C        lea eax,dword ptr ss:[ebp+C]
     75D0E3A2   50             push eax
     75D0E3A3   FF15 E011D075  call dword ptr ds:[<&RtlGetCurrentProce
     75D0E3A9   56             push esi

<kernelbase.VirtualAllocEx>

.text:75D0E378 kernelbase.dll:$E378 #D778
```

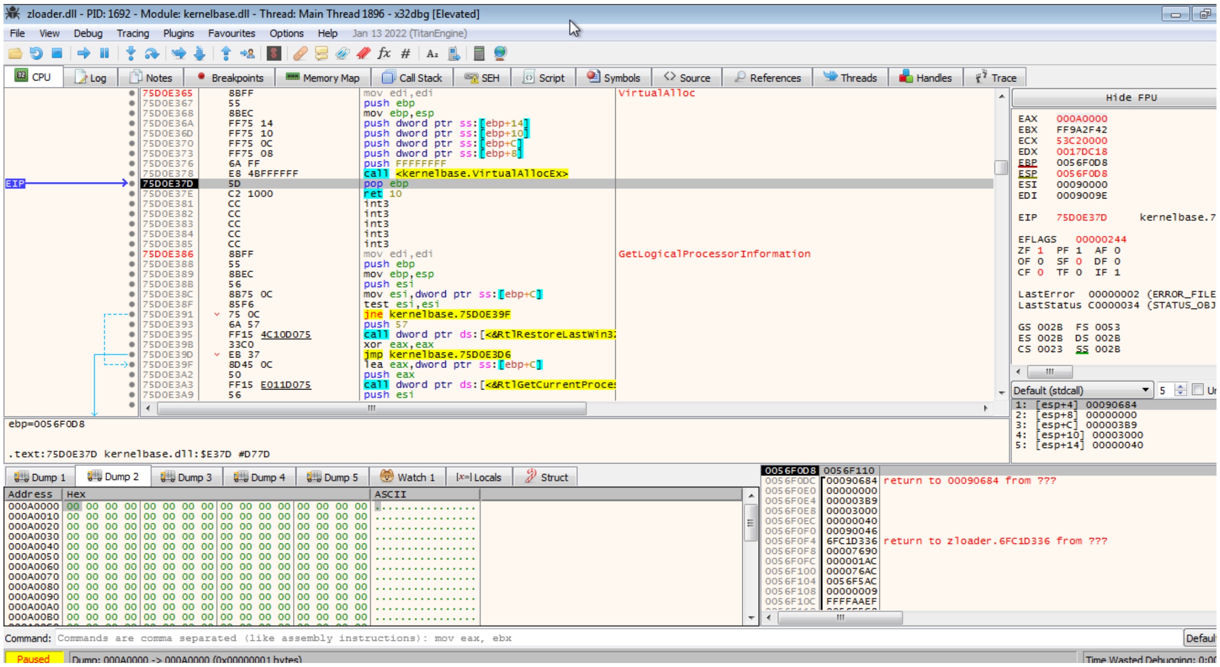
```
EAX   6FBE0000     zloader.6FBE00
EBX   FF9A2F42
ECX   70B5F000
EDX   00090046
EBP   0056F0D8
ESP   0056F0C4
ESI   00090000
EDI   0009009E

EIP   75D0E378     kernelbase.75D

EFLAGS   00000287
ZF 0  PF 1  AF 0
OF 0  SF 1  DF 0
CF 1  TF 0  IF 1

LastError  00000002 (ERROR_FILE_N
LastStatus C0000034 (STATUS_OBJEC

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

1: [esp] FFFFFFFF
2: [esp+4] 00000000
3: [esp+8] 000003B9
4: [esp+C] 00003000
5: [esp+10] 00000040
```

```
Address   Hex                                              ASCII
00090000  90 E8 00 00 00 00 5B 8D 43 3C B9 CE 01 00 00 BF  .è....[.C<'Î...¿
00090010  8C 17 C6 0B 81 C1 70 02 00 00 B9 BA 09 00 00 89  ..Æ..Áp...'°....
00090020  FA 31 DB 89 CE 83 E6 03 75 0E 89 FB 66 01 DA 6B  ú1Û.Î.æ.u..ûf.Úk
00090030  D2 03 90 C1 CA 08 89 D7 30 10 40 E2 E6 E9 43 05  Ò..ÁÊ..x0.@âæéC.
00090040  00 00 FF FF FF FF 02 0A 00 00 36 87 03 00 00 DE  ..ÿÿÿÿ....6....Þ
00090050  02 00 99 B8 01 00 78 13 06 00 B9 03 00 00 12 03  ......x...'.....
00090060  89 81 14 06 11 05 03 0A 03 02 03 04 02 0F 00 00  ................
00090070  0F 19 03 05 04 02 07 02 00 00 54 34 06 00 00 00  ..........T4....
00090080  00 00 06 00 00 00 8E 18 07 00 3C D1 38 00 AE 68  ..........<Ñ8.®h
00090090  00 00 FA 8B 34 00 DA 60 35 00 42 31 0E 00 55 89  ..ú.4.Ú`5.B1..U.
000900A0  E5 51 56 57 8B 4D 0C 8B 75 10 8B 7D 14 FF 36 FF  åQVW.M...u..}.ÿ6ÿ
000900B0  75 08 E8 13 00 00 00 89 07 83 C7 04 83 C6 04 E2  u.è.......Ç..Æ.â
```

```
0056F0C4  FFFFFFFF
0056F0C8  00000000
0056F0CC  000003B9
0056F0D0  00003000
0056F0D4  00000040
0056F0D8  0056F110
0056F0DC  00090684  return to 00090684 from ???
0056F0E0  00000000
0056F0E4  000003B9
0056F0E8  00003000
0056F0EC  00000040
0056F0F0  00090046
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8  00007690
```

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

The address returned by it is 40000000 and we dumped it in dump2.


zloader.dll - PID: 1692 - Module: kernelbase.dll - Thread: Main Thread 1896 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
Hide FPU
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct
Default (stdcall) 5 Ur
Defaul
Paused
Time Wasted Debugging: 0:0

```
     75D0E365   8BFF           mov edi,edi                              VirtualAlloc
     75D0E367   55             push ebp
     75D0E368   8BEC           mov ebp,esp
     75D0E36A   FF75 14        push dword ptr ss:[ebp+14]
     75D0E36D   FF75 10        push dword ptr ss:[ebp+10]
     75D0E370   FF75 0C        push dword ptr ss:[ebp+C]
     75D0E373   FF75 08        push dword ptr ss:[ebp+8]
     75D0E376   6A FF          push FFFFFFFF
     75D0E378   E8 4BFFFFFF    call <kernelbase.VirtualAllocEx>
EIP  75D0E37D   5D             pop ebp
     75D0E37E   C2 1000        ret 10
     75D0E381   CC             int3
     75D0E382   CC             int3
     75D0E383   CC             int3
     75D0E384   CC             int3
     75D0E385   CC             int3
     75D0E386   8BFF           mov edi,edi                              GetLogicalProcessorInformation
     75D0E388   55             push ebp
     75D0E389   8BEC           mov ebp,esp
     75D0E38B   56             push esi
     75D0E38C   8B75 0C        mov esi,dword ptr ss:[ebp+C]
     75D0E38F   85F6           test esi,esi
     75D0E391   75 0C          jne kernelbase.75D0E39F
     75D0E393   6A 57          push 57
     75D0E395   FF15 4C10D075  call dword ptr ds:[<&RtlRestoreLastWin3
     75D0E39B   33C0           xor eax,eax
     75D0E39D   EB 37          jmp kernelbase.75D0E3D6
     75D0E39F   8D45 0C        lea eax,dword ptr ss:[ebp+C]
     75D0E3A2   50             push eax
     75D0E3A3   FF15 E011D075  call dword ptr ds:[<&RtlGetCurrentProce
     75D0E3A9   56             push esi

ebp=0056F0D8

.text:75D0E37D kernelbase.dll:$E37D #D77D
```

```
EAX   000A0000
EBX   FF9A2F42
ECX   53C20000
EDX   0017DC18
EBP   0056F0D8
ESP   0056F0D8
ESI   00090000
EDI   0009009E

EIP   75D0E37D     kernelbase.7

EFLAGS   00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000002 (ERROR_FILE
LastStatus C0000034 (STATUS_OBJ

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

1: [esp+4] 00090684
2: [esp+8] 00000000
3: [esp+C] 000003B9
4: [esp+10] 00003000
5: [esp+14] 00000040
```

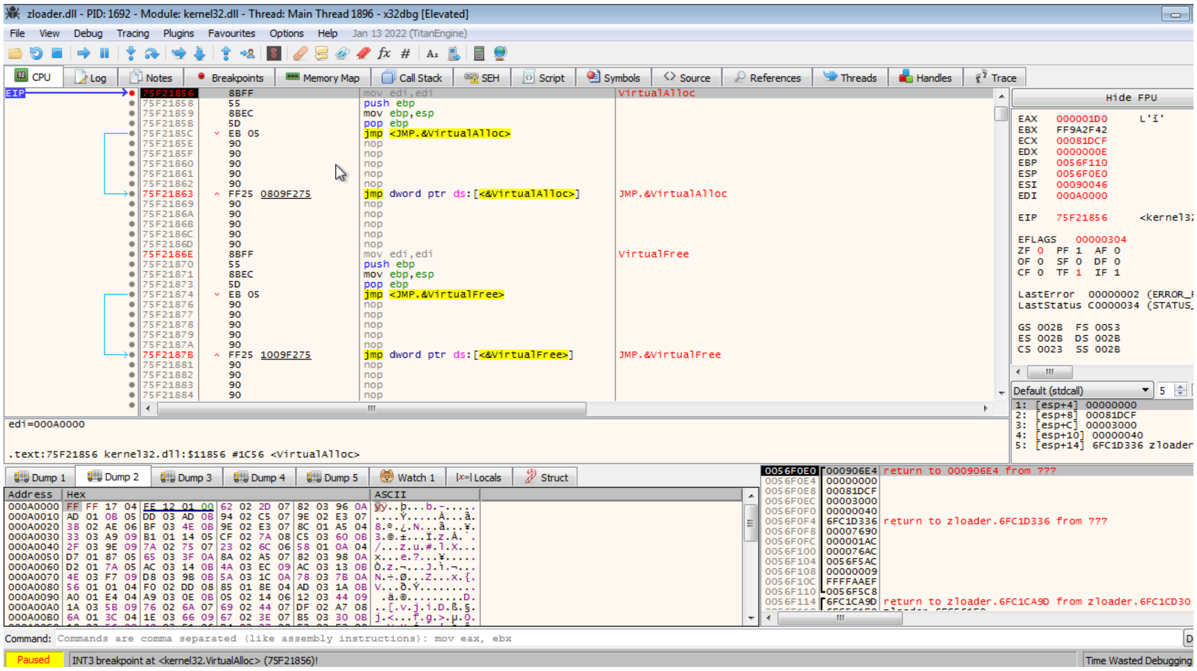
```
Address   Hex                                              ASCII
000A0000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A0090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000A00B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
0056F0D8  0056F110
0056F0DC  00090684  return to 00090684 from ???
0056F0E0  00000000
0056F0E4  000003B9
0056F0E8  00003000
0056F0EC  00000040
0056F0F0  00090046
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8  00007690
0056F0FC  000001AC
0056F100  000076AC
0056F104  0056F5AC
0056F108  00000009
0056F10C  FFFFAAEF
```

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

Dump: 000A0000 -> 000A0000 (0x00000001 bytes)

Now here again we hit VirtualAlloc and our dump2 is filled with some values just step over .



zloader.dll - PID: 1692 - Module: kernel32.dll - Thread: Main Thread 1896 - x32dbg [Elevated]

```
EIP → 75F21856    8BFF           mov edi,edi                        VirtualAlloc
      75F21858    55             push ebp
      75F21859    8BEC           mov ebp,esp
      75F2185B    5D             pop ebp
      75F2185C    EB 05          jmp <JMP.&VirtualAlloc>
      75F2185E    90             nop
      75F2185F    90             nop
      75F21860    90             nop
      75F21861    90             nop
      75F21862    90             nop
      75F21863    FF25 0809F275  jmp dword ptr ds:[<&VirtualAlloc>] JMP.&VirtualAlloc
      75F21869    90             nop
      75F2186A    90             nop
      75F2186B    90             nop
      75F2186C    90             nop
      75F2186D    90             nop
      75F2186E    8BFF           mov edi,edi                        VirtualFree
      75F21870    55             push ebp
      75F21871    8BEC           mov ebp,esp
      75F21873    5D             pop ebp
      75F21874    EB 05          jmp <JMP.&VirtualFree>
      75F21876    90             nop
      75F21877    90             nop
      75F21878    90             nop
      75F21879    90             nop
      75F2187A    90             nop
      75F2187B    FF25 1009F275  jmp dword ptr ds:[<&VirtualFree>]  JMP.&VirtualFree
      75F21881    90             nop
      75F21882    90             nop
      75F21883    90             nop
      75F21884    90             nop

edi=000A0000

.text:75F21856 kernel32.dll:$11856 #1C56 <VirtualAlloc>
```

```
EAX     000001D0     L'ǐ'
EBX     FF9A2F42
ECX     00081DCF
EDX     0000000E
EBP     0056F110
ESP     0056F0E0
ESI     00090046
EDI     000A0000

EIP     75F21856     <kernel3

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

LastError  00000002 (ERROR_F
LastStatus C0000034 (STATUS_

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

1: [esp+4] 00000000
2: [esp+8] 00081DCF
3: [esp+C] 00003000
4: [esp+10] 00000040
5: [esp+14] 6FC1D336 zloader
```

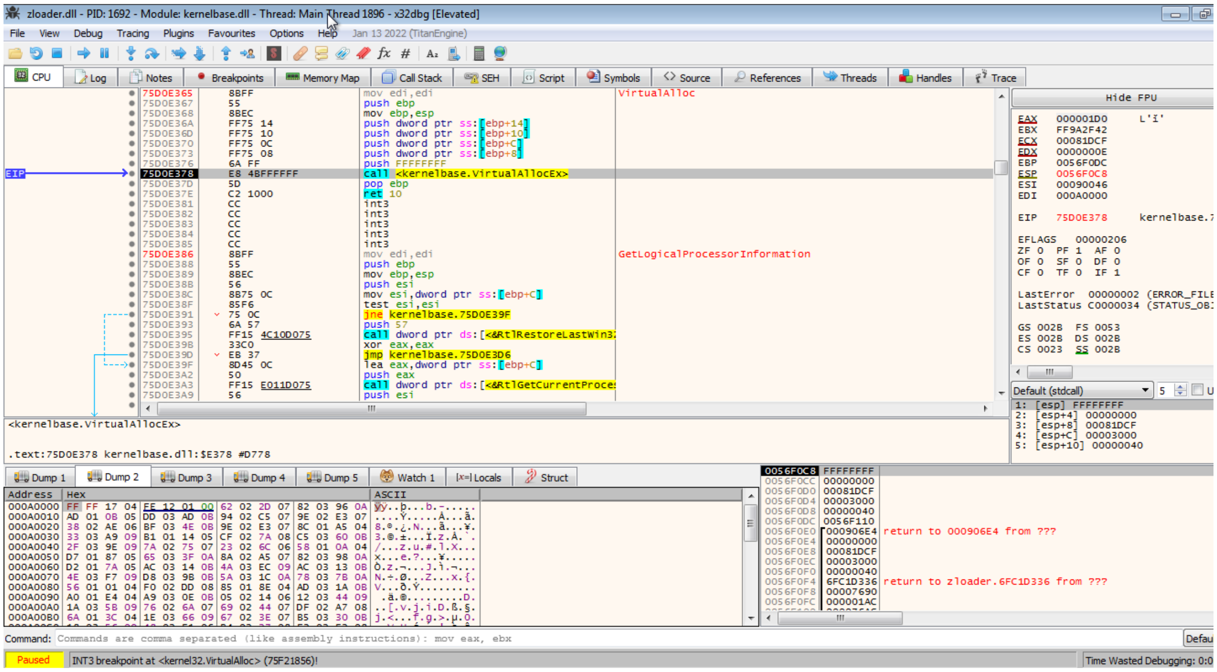
```
Address   Hex                                              ASCII
000A0000  FF FF 17 04 FE 12 01 00 62 02 2D 07 82 03 96 0A  ÿÿ..þ...b.-.....
000A0010  AD 01 0B 05 DD 03 AD 0B 94 02 C5 07 9E 02 E3 07  ....Ý.....Å...ã.
000A0020  38 02 AE 06 BF 03 4E 0B 9E 02 E3 07 8C 01 A5 04  8.®.¿.N...ã...¥.
000A0030  33 03 A9 09 B1 01 14 05 CF 02 7A 08 C5 03 60 0B  3.©.±...Ï.z.Å.`.
000A0040  2F 03 9E 09 7A 02 75 07 23 02 6C 06 58 01 0A 04  /...z.u.#.l.X...
000A0050  D7 01 87 05 65 03 3F 0A 8A 02 A5 07 82 03 98 0A  ×...e.?...¥.....
000A0060  D2 01 7A 05 AC 03 14 0B 4A 03 EC 09 AC 03 13 0B  Ò.z.¬...J.ì.¬...
000A0070  4E 03 F7 09 D8 03 9B 0B 5A 03 1C 0A 78 03 7B 0A  N.÷.Ø...Z...x.{.
000A0080  56 01 01 04 F0 02 DD 08 85 01 8E 04 AD 03 1A 0B  V...ð.Ý.........
000A0090  A0 01 E4 04 A9 03 0E 0B 05 02 14 06 12 03 44 09   .ä.©.........D.
000A00A0  1A 03 5B 09 76 02 6A 07 69 02 44 07 DF 02 A7 08  ..[.v.j.i.D.ß.§.
000A00B0  6A 01 3C 04 1E 03 66 09 67 02 3E 07 B5 03 30 0B  j.<...f.g.>.µ.0.
```

```
0056F0E0  000906E4  return to 000906E4 from ???
0056F0E4  00000000
0056F0E8  00081DCF
0056F0EC  00003000
0056F0F0  00000040
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8  00007690
0056F0FC  000001AC
0056F100  000076AC
0056F104  0056F5AC
0056F108  00000009
0056F10C  FFFFAAEF
0056F110  0056F5C8
0056F114  6FC1CA9D  return to zloader.6FC1CA9D from zloader.6FC1CD30
```

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

Paused   INT3 breakpoint at <kernel32.VirtualAlloc> (75F21856)!   Time Wasted Debugging

Here just step over it and dump the address return by it in eax.



zloader.dll - PID: 1692 - Module: kernelbase.dll - Thread: Main Thread 1896 - x32dbg [Elevated]

```
      75D0E365    8BFF           mov edi,edi                        VirtualAlloc
      75D0E367    55             push ebp
      75D0E368    8BEC           mov ebp,esp
      75D0E36A    FF75 14        push dword ptr ss:[ebp+14]
      75D0E36D    FF75 10        push dword ptr ss:[ebp+10]
      75D0E370    FF75 0C        push dword ptr ss:[ebp+C]
      75D0E373    FF75 08        push dword ptr ss:[ebp+8]
      75D0E376    6A FF          push FFFFFFFF
EIP → 75D0E378    E8 4BFFFFFF    call <kernelbase.VirtualAllocEx>
      75D0E37D    5D             pop ebp
      75D0E37E    C2 1000        ret 10
      75D0E381    CC             int3
      75D0E382    CC             int3
      75D0E383    CC             int3
      75D0E384    CC             int3
      75D0E385    CC             int3
      75D0E386    8BFF           mov edi,edi                        GetLogicalProcessorInformation
      75D0E388    55             push ebp
      75D0E389    8BEC           mov ebp,esp
      75D0E38B    56             push esi
      75D0E38C    8B75 0C        mov esi,dword ptr ss:[ebp+C]
      75D0E38F    85F6           test esi,esi
      75D0E391    75 0C          jne kernelbase.75D0E39F
      75D0E393    6A 57          push 57
      75D0E395    FF15 4C10D075  call dword ptr ds:[<&RtlRestoreLastWin32
      75D0E39B    33C0           xor eax,eax
      75D0E39D    EB 37          jmp kernelbase.75D0E3D6
      75D0E39F    8D45 0C        lea eax,dword ptr ss:[ebp+C]
      75D0E3A2    50             push eax
      75D0E3A3    FF15 E011D075  call dword ptr ds:[<&RtlGetCurrentProces
      75D0E3A9    56             push esi

<kernelbase.VirtualAllocEx>

.text:75D0E378 kernelbase.dll:$E378 #D778
```

```
EAX     000001D0     L'ǐ'
EBX     FF9A2F42
ECX     00081DCF
EDX     0000000E
EBP     0056F0DC
ESP     0056F0C8
ESI     00090046
EDI     000A0000

EIP     75D0E378     kernelbase.7

EFLAGS   00000206
ZF 0   PF 1   AF 0
OF 0   SF 0   DF 0
CF 0   TF 0   IF 1

LastError  00000002 (ERROR_FILE
LastStatus C0000034 (STATUS_OBJ

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

1: [esp] FFFFFFFF
2: [esp+4] 00000000
3: [esp+8] 00081DCF
4: [esp+C] 00003000
5: [esp+10] 00000040
```

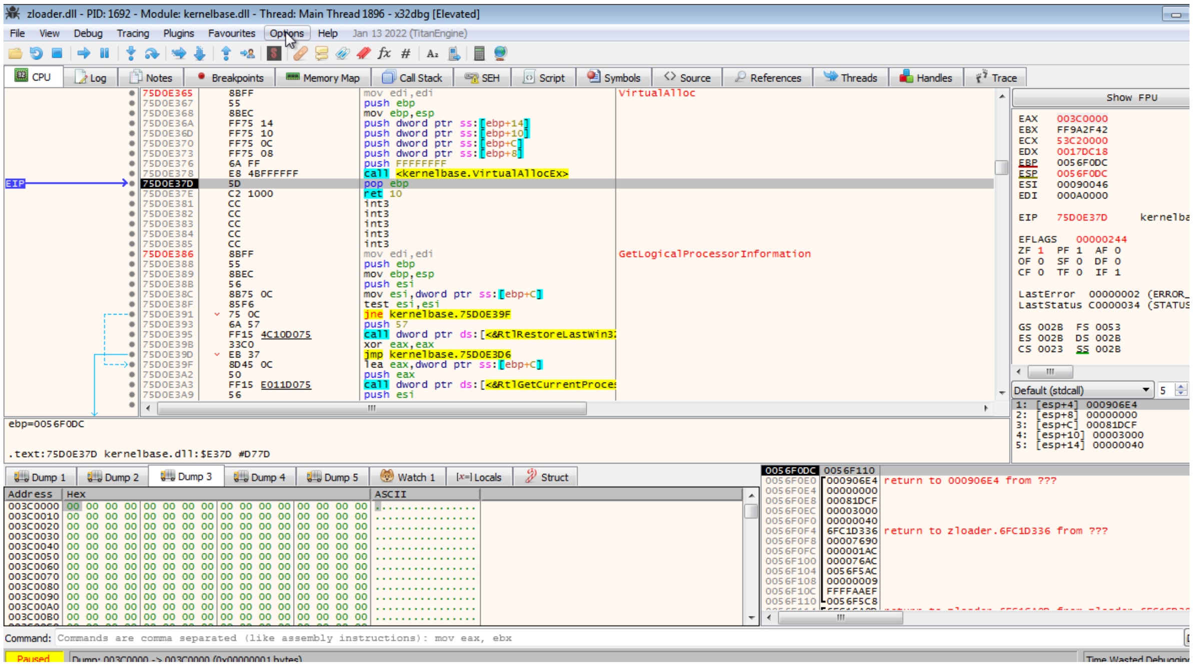
```
Address   Hex                                              ASCII
000A0000  FF FF 17 04 FE 12 01 00 62 02 2D 07 82 03 96 0A  ÿÿ..þ...b.-.....
000A0010  AD 01 0B 05 DD 03 AD 0B 94 02 C5 07 9E 02 E3 07  ....Ý.....Å...ã.
000A0020  38 02 AE 06 BF 03 4E 0B 9E 02 E3 07 8C 01 A5 04  8.®.¿.N...ã...¥.
000A0030  33 03 A9 09 B1 01 14 05 CF 02 7A 08 C5 03 60 0B  3.©.±...Ï.z.Å.`.
000A0040  2F 03 9E 09 7A 02 75 07 23 02 6C 06 58 01 0A 04  /...z.u.#.l.X...
000A0050  D7 01 87 05 65 03 3F 0A 8A 02 A5 07 82 03 98 0A  ×...e.?...¥.....
000A0060  D2 01 7A 05 AC 03 14 0B 4A 03 EC 09 AC 03 13 0B  Ò.z.¬...J.ì.¬...
000A0070  4E 03 F7 09 D8 03 9B 0B 5A 03 1C 0A 78 03 7B 0A  N.÷.Ø...Z...x.{.
000A0080  56 01 01 04 F0 02 DD 08 85 01 8E 04 AD 03 1A 0B  V...ð.Ý.........
000A0090  A0 01 E4 04 A9 03 0E 0B 05 02 14 06 12 03 44 09   .ä.©.........D.
000A00A0  1A 03 5B 09 76 02 6A 07 69 02 44 07 DF 02 A7 08  ..[.v.j.i.D.ß.§.
000A00B0  6A 01 3C 04 1E 03 66 09 67 02 3E 07 B5 03 30 0B  j.<...f.g.>.µ.0.
```

```
0056F0C8  FFFFFFFF
0056F0CC  00000000
0056F0D0  00081DCF
0056F0D4  00003000
0056F0D8  00000040
0056F0DC  0056F110
0056F0E0  000906E4  return to 000906E4 from ???
0056F0E4  00000000
0056F0E8  00081DCF
0056F0EC  00003000
0056F0F0  00000040
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8  00007690
0056F0FC  000001AC
```

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

Paused   INT3 breakpoint at <kernel32.VirtualAlloc> (75F21856)!   Time Wasted Debugging: 0:0

Here the address return is 3C0000, and we dump it in dump3 and press f9.


zloader.dll - PID: 1692 - Module: kernelbase.dll - Thread: Main Thread 1896 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct
Show FPU
Default (stdcall)

```
75D0E365   8BFF          mov edi,edi                                   VirtualAlloc
75D0E367   55            push ebp
75D0E368   8BEC          mov ebp,esp
75D0E36A   FF75 14       push dword ptr ss:[ebp+14]
75D0E36D   FF75 10       push dword ptr ss:[ebp+10]
75D0E370   FF75 0C       push dword ptr ss:[ebp+C]
75D0E373   FF75 08       push dword ptr ss:[ebp+8]
75D0E376   6A FF         push FFFFFFFF
75D0E378   E8 4BFFFFFF   call <kernelbase.VirtualAllocEx>
EIP → 75D0E37D   5D            pop ebp
75D0E37E   C2 1000       ret 10
75D0E381   CC            int3
75D0E382   CC            int3
75D0E383   CC            int3
75D0E384   CC            int3
75D0E385   CC            int3
75D0E386   8BFF          mov edi,edi                                   GetLogicalProcessorInformation
75D0E388   55            push ebp
75D0E389   8BEC          mov ebp,esp
75D0E38B   56            push esi
75D0E38C   8B75 0C       mov esi,dword ptr ss:[ebp+C]
75D0E38F   85F6          test esi,esi
75D0E391   75 0C         jne kernelbase.75D0E39F
75D0E393   6A 57         push 57
75D0E395   FF15 4C10D075 call dword ptr ds:[<&RtlRestoreLastWin3
75D0E39B   33C0          xor eax,eax
75D0E39D   EB 37         jmp kernelbase.75D0E3D6
75D0E39F   8D45 0C       lea eax,dword ptr ss:[ebp+C]
75D0E3A2   50            push eax
75D0E3A3   FF15 E011D075 call dword ptr ds:[<&RtlGetCurrentProce
75D0E3A9   56            push esi

ebp=0056F0DC

.text:75D0E37D kernelbase.dll:$E37D #D77D
```

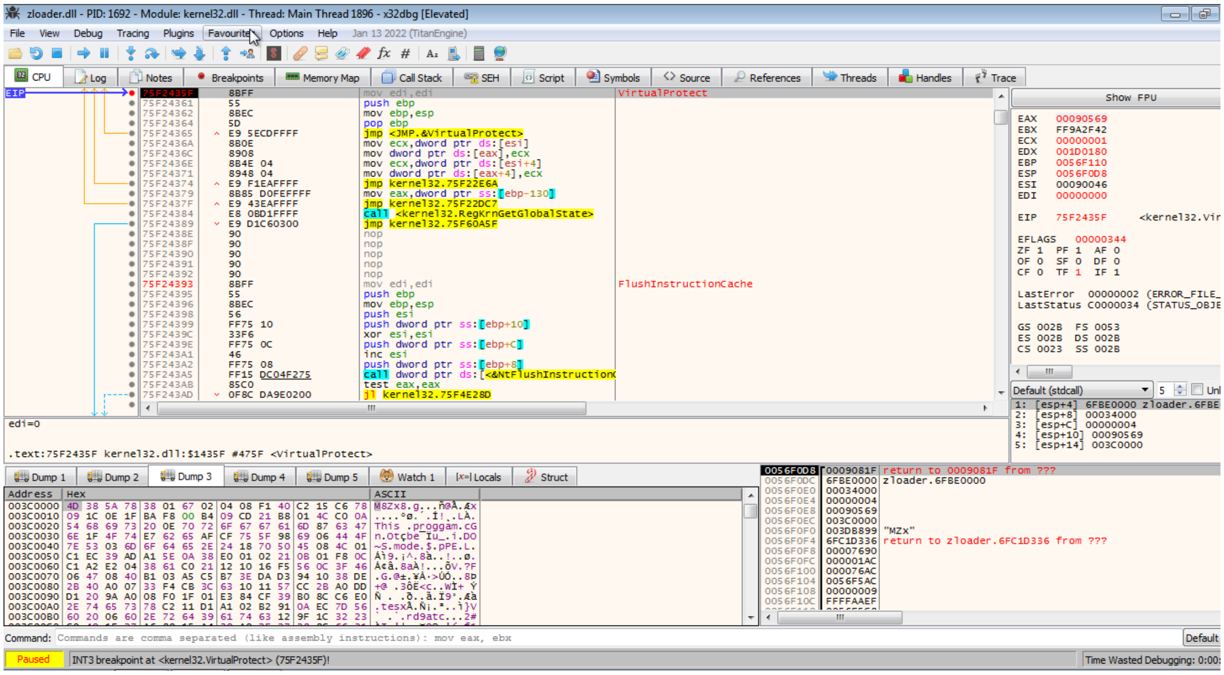
```
EAX   003C0000
EBX   FF9A2F42
ECX   53C20000
EDX   0017DC18
EBP   0056F0DC
ESP   0056F0DC
ESI   00090046
EDI   000A0000

EIP   75D0E37D     kernelba

EFLAGS   00000244
ZF 1   PF 1   AF 0
OF 0   SF 0   DF 0
CF 0   TF 0   IF 1

LastError  00000002 (ERROR_
LastStatus C0000034 (STATUS

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

Default (stdcall)  5
1: [esp+4] 000906E4
2: [esp+8] 00000000
3: [esp+C] 00081DCF
4: [esp+10] 00003000
5: [esp+14] 00000040
```

```
Address   Hex                                              ASCII
003C0000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003C0010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003C0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003C0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003C0040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003C0050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003C0060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003C0070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003C0080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003C0090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003C00A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003C00B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
0056F0DC  0056F110
0056F0E0  000906E4  return to 000906E4 from ???
0056F0E4  00000000
0056F0E8  00081DCF
0056F0EC  00003000
0056F0F0  00000040
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8  00007690
0056F0FC  000001AC
0056F100  000076AC
0056F104  0056F5AC
0056F108  00000009
0056F10C  FFFFAAEF
0056F110  0056F5C8
```

```
Command: Commands are comma separated (like assembly instructions): mov eax, ebx
Paused  Dump: 003C0000 -> 003C0000 (0x00000001 bytes)  Time Wasted Debuggin
```

Here we get breakpoint at VirtualProtect just step over it and note the dump3 is filled with exe.


zloader.dll - PID: 1692 - Module: kernel32.dll - Thread: Main Thread 1896 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct
Show FPU
Default (stdcall)
Default

```
EIP → 75F2435F   8BFF          mov edi,edi                             VirtualProtect
75F24361   55            push ebp
75F24362   8BEC          mov ebp,esp
75F24364   5D            pop ebp
75F24365   E9 5ECDFFFF   jmp <JMP.&VirtualProtect>
75F2436A   8B0E          mov ecx,dword ptr ds:[esi]
75F2436C   8908          mov dword ptr ds:[eax],ecx
75F2436E   8B4E 04       mov ecx,dword ptr ds:[esi+4]
75F24371   8948 04       mov dword ptr ds:[eax+4],ecx
75F24374   E9 F1EAFFFF   jmp kernel32.75F22E6A
75F24379   8B85 D0FEFFFF mov eax,dword ptr ss:[ebp-130]
75F2437F   E9 43EAFFFF   jmp kernel32.75F22DC7
75F24384   E8 0BD1FFFF   call <kernel32.RegKrnGetGlobalState>
75F24389   E9 D1C60300   jmp kernel32.75F60A5F
75F2438E   90            nop
75F2438F   90            nop
75F24390   90            nop
75F24391   90            nop
75F24392   90            nop
75F24393   8BFF          mov edi,edi                             FlushInstructionCache
75F24395   55            push ebp
75F24396   8BEC          mov ebp,esp
75F24398   56            push esi
75F24399   FF75 10       push dword ptr ss:[ebp+10]
75F2439C   33F6          xor esi,esi
75F2439E   FF75 0C       push dword ptr ss:[ebp+C]
75F243A1   46            inc esi
75F243A2   FF75 08       push dword ptr ss:[ebp+8]
75F243A5   FF15 DC04F275 call dword ptr ds:[<&NtFlushInstruction
75F243AB   85C0          test eax,eax
75F243AD   0F8C DA9E0200 jl kernel32.75F4E28D

edi=0

.text:75F2435F kernel32.dll:$1435F #475F <VirtualProtect>
```

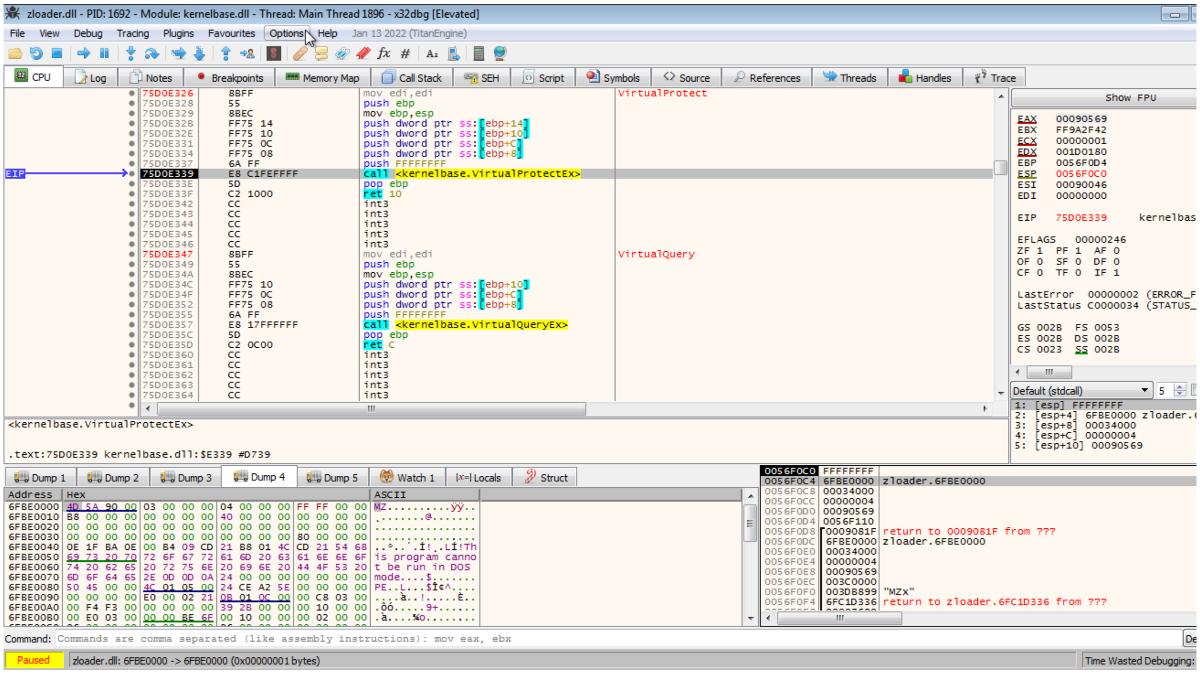
```
EAX   00090569
EBX   FF9A2F42
ECX   00000001
EDX   001D0180
EBP   0056F110
ESP   0056F0D8
ESI   00090046
EDI   00000000

EIP   75F2435F     <kernel32.Vir

EFLAGS   00000344
ZF 1   PF 1   AF 0
OF 0   SF 0   DF 0
CF 0   TF 1   IF 1

LastError  00000002 (ERROR_FILE_
LastStatus C0000034 (STATUS_OBJE

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

Default (stdcall)  5  Unl
1: [esp+4] 6FBE0000 zloader.6FBE
2: [esp+8] 00034000
3: [esp+C] 00000004
4: [esp+10] 00090569
5: [esp+14] 003C0000
```

```
Address   Hex                                              ASCII
003C0000  4D 38 5A 78 38 01 67 02 04 08 F1 40 C2 15 C6 78  M8Zx8.g...ñ@Â.Æx
003C0010  09 1C 0E 1F BA F8 00 B4 09 CD 21 B8 01 4C C0 0A  ....ºø.´.Í!¸.LÀ.
003C0020  54 68 69 73 20 0E 70 72 6F 67 67 61 6D 87 63 47  This .proggam.cG
003C0030  6E 1F 4F 74 E7 62 65 AF CF 75 5F 98 69 06 44 4F  n.Otçbe¯Ïu_.i.DO
003C0040  7E 53 03 6D 6F 64 65 2E 24 18 70 50 45 08 4C 01  ~S.mode.$.pPE.L.
003C0050  C1 EC 39 AD A1 5E 0A 38 E0 01 02 21 0B 01 F8 0C  Áì9.¡^.8à..!..ø.
003C0060  C1 A2 E2 04 38 61 C0 21 12 10 16 F5 56 0C 3F 46  Á¢â.8aÀ!...õV.?F
003C0070  06 47 08 40 B1 03 A5 C5 B7 3E DA D3 94 10 38 DE  .G.@±.¥Å·>ÚÓ..8Þ
003C0080  2B 40 A0 07 33 F4 CB 3C 63 10 11 57 CC 2B A0 DD  +@ .3ôË<c..WÌ+ Ý
003C0090  D1 20 9A A0 08 F0 1F 01 E3 84 CF 39 B0 8C C6 E0  Ñ . .ð..ã.Ï9°.Æà
003C00A0  2E 74 65 73 78 C2 11 D1 A1 02 B2 91 0A EC 7D 56  .tesxÂ.Ñ¡.²..ì}V
003C00B0  60 20 06 60 2E 72 64 39 61 74 63 12 9F 1C 32 23  ` .`.rd9atc...2#
```

```
0056F0D8  0009081F  return to 0009081F from ???
0056F0DC  6FBE0000  zloader.6FBE0000
0056F0E0  00034000
0056F0E4  00000004
0056F0E8  00090569
0056F0EC  003C0000
0056F0F0  003DB899  "MZx"
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8  00007690
0056F0FC  000001AC
0056F100  000076AC
0056F104  0056F5AC
0056F108  00000009
0056F10C  FFFFAAEF
```

```
Command: Commands are comma separated (like assembly instructions): mov eax, ebx
Paused  INT3 breakpoint at <kernel32.VirtualProtect> (75F2435F)!  Time Wasted Debugging: 0:00:
```

Here just step over it and note the second parameter to it i.e. 6FBE0000.



zloader.dll - PID: 1692 - Module: kernelbase.dll - Thread: Main Thread 1896 - x32dbg [Elevated]

```
75D0E326    8BFF            mov edi,edi                               VirtualProtect
75D0E328    55              push ebp
75D0E329    8BEC            mov ebp,esp
75D0E32B    FF75 14         push dword ptr ss:[ebp+14]
75D0E32E    FF75 10         push dword ptr ss:[ebp+10]
75D0E331    FF75 0C         push dword ptr ss:[ebp+C]
75D0E334    FF75 08         push dword ptr ss:[ebp+8]
75D0E337    6A FF           push FFFFFFFF
75D0E339    E8 C1FEFFFF     call <kernelbase.VirtualProtectEx>
75D0E33E    5D              pop ebp
75D0E33F    C2 1000         ret 10
75D0E342    CC              int3
75D0E343    CC              int3
75D0E344    CC              int3
75D0E345    CC              int3
75D0E346    CC              int3
75D0E347    8BFF            mov edi,edi                               VirtualQuery
75D0E349    55              push ebp
75D0E34A    8BEC            mov ebp,esp
75D0E34C    FF75 10         push dword ptr ss:[ebp+10]
75D0E34F    FF75 0C         push dword ptr ss:[ebp+C]
75D0E352    FF75 08         push dword ptr ss:[ebp+8]
75D0E355    6A FF           push FFFFFFFF
75D0E357    E8 17FFFFFF     call <kernelbase.VirtualQueryEx>
75D0E35C    5D              pop ebp
75D0E35D    C2 0C00         ret C
75D0E360    CC              int3
75D0E361    CC              int3
75D0E362    CC              int3
75D0E363    CC              int3
75D0E364    CC              int3

<kernelbase.VirtualProtectEx>

.text:75D0E339 kernelbase.dll:$E339 #D739
```

```
EAX     00090569
EBX     FF9A2F42
ECX     00000001
EDX     001D0180
EBP     0056F0D4
ESP     0056F0C0
ESI     00090046
EDI     00000000

EIP     75D0E339     kernelbas

EFLAGS    00000246
ZF 1   PF 1   AF 0
OF 0   SF 0   DF 0
CF 0   TF 0   IF 1

LastError   00000002 (ERROR_F
LastStatus  C0000034 (STATUS_

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

Default (stdcall)   5
1: [esp] FFFFFFFF
2: [esp+4] 6FBE0000 zloader.(
3: [esp+8] 00034000
4: [esp+C] 00000004
5: [esp+10] 00090569
```

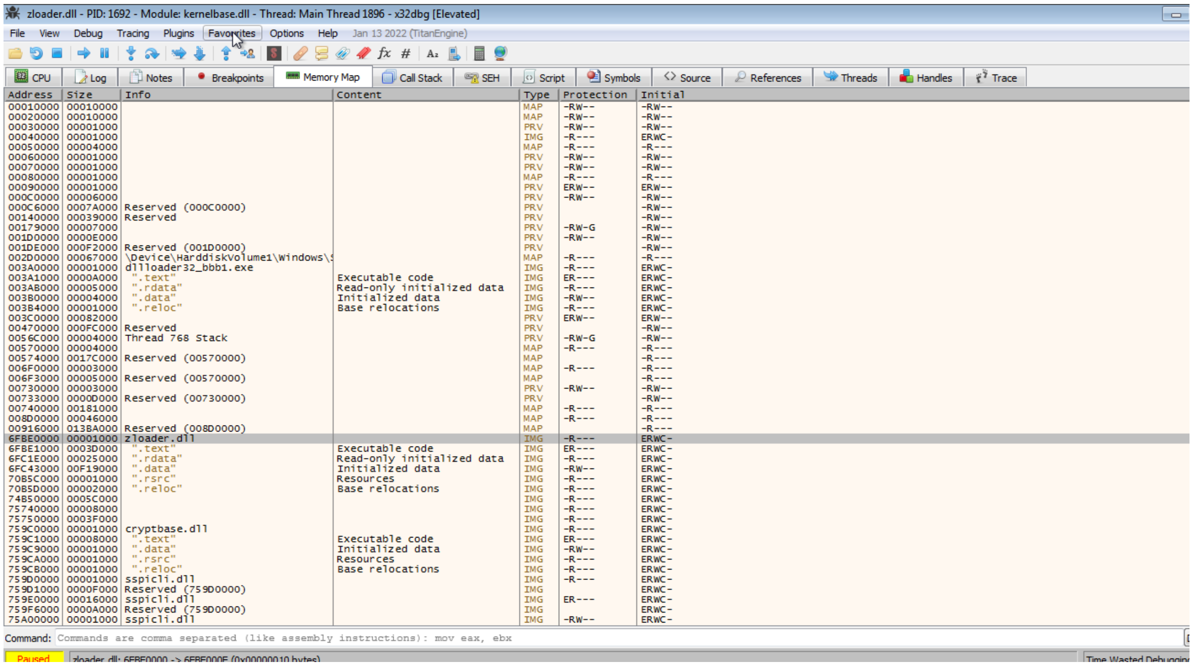
```
Address   Hex                                              ASCII
6FBE0000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
6FBE0010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
6FBE0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
6FBE0030  00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00  ................
6FBE0040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
6FBE0050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
6FBE0060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
6FBE0070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
6FBE0080  50 45 00 00 4C 01 05 00 24 CE A2 5E 00 00 00 00  PE..L...$Î¢^....
6FBE0090  00 00 00 00 E0 00 02 21 0B 01 0C 00 00 C8 03 00  ....à..!.....È..
6FBE00A0  00 F4 F3 00 00 00 00 00 39 2B 00 00 00 10 00 00  .ôó.....9+......
6FBE00B0  00 E0 03 00 00 00 BE 6F 00 10 00 00 00 02 00 00  .à....¾o........
```

```
0056F0C0  FFFFFFFF
0056F0C4  6FBE0000  zloader.6FBE0000
0056F0C8  00034000
0056F0CC  00000004
0056F0D0  00090569
0056F0D4  0056F110
0056F0D8  0009081F  return to 0009081F from ???
0056F0DC  6FBE0000  zloader.6FBE0000
0056F0E0  00034000
0056F0E4  00000004
0056F0E8  00090569
0056F0EC  003C0000
0056F0F0  003DB899  "MZx"
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
```

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

Paused zloader.dll: 6FBE0000 -> 6FBE0000 (0x00000001 bytes) Time Wasted Debugging:

We view that memory region 6FBE0000 in Memory Map , here before the call we see that memory region has read only permission.



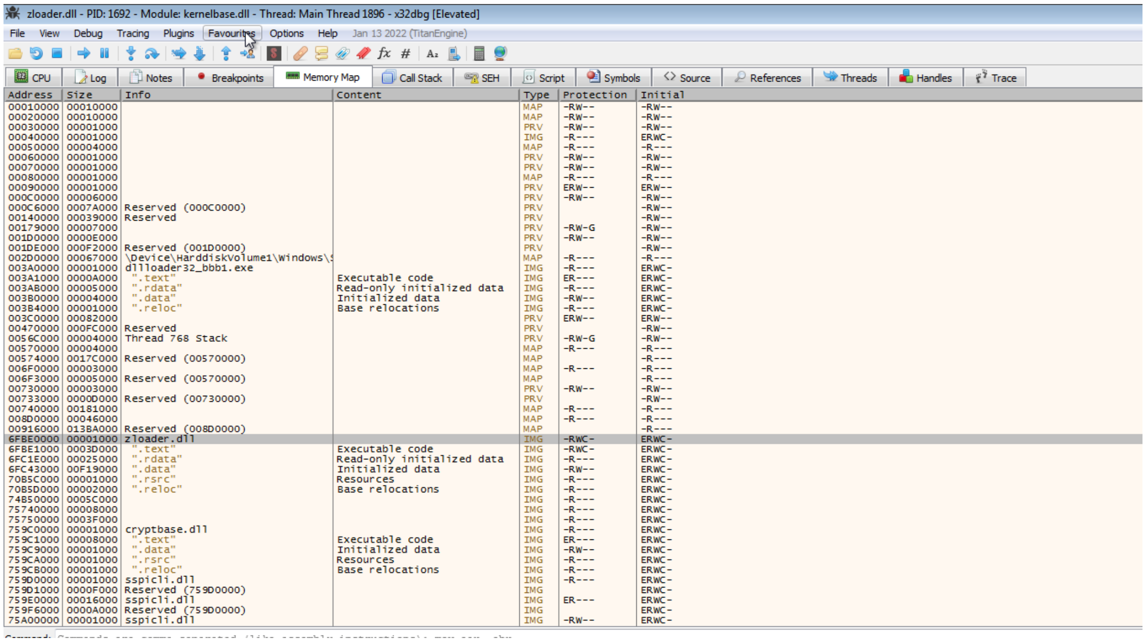
zloader.dll - PID: 1692 - Module: kernelbase.dll - Thread: Main Thread 1896 - x32dbg [Elevated]

Address	Size	Info	Content	Type	Protection	Initial
00010000	00010000			MAP	-RW--	-RW--
00020000	00010000			MAP	-RW--	-RW--
00030000	00001000			PRV	-RW--	-RW--
00040000	00001000			IMG	-R---	ERWC-
00050000	00004000			MAP	-R---	-R---
00060000	00001000			PRV	-RW--	-RW--
00070000	00001000			PRV	-RW--	-RW--
00080000	00001000			MAP	-R---	-R---
00090000	00001000			PRV	ERW--	ERW--
000C0000	00006000			PRV	-RW--	-RW--
000C6000	0007A000	Reserved (000C0000)		PRV		-RW--
00140000	00039000	Reserved		PRV		-RW--
00179000	00007000			PRV	-RW-G	-RW--
001D0000	0000E000			PRV	-RW--	-RW--
001DE000	000F2000	Reserved (001D0000)		PRV		-RW--
002D0000	00067000	\Device\HarddiskVolume1\Windows\S		MAP	-R---	-R---
003A0000	00001000	dllloader32_bbb1.exe		IMG	-R---	ERWC-
003A1000	0000A000	".text"	Executable code	IMG	ER---	ERWC-
003AB000	00005000	".rdata"	Read-only initialized data	IMG	-R---	ERWC-
003B0000	00004000	".data"	Initialized data	IMG	-RW--	ERWC-
003B4000	00001000	".reloc"	Base relocations	IMG	-R---	ERWC-
003C0000	00082000			PRV	ERW--	ERW--
00470000	000FC000	Reserved		PRV		-RW--
0056C000	00004000	Thread 768 Stack		PRV	-RW-G	-RW--
00570000	00004000			MAP	-R---	-R---
00574000	0017C000	Reserved (00570000)		MAP		-R---
006F0000	00003000			MAP	-R---	-R---
006F3000	00005000	Reserved (00570000)		MAP		-R---
00730000	00003000			PRV	-RW--	-RW--
00733000	0000D000	Reserved (00730000)		PRV		-RW--
00740000	00181000			MAP	-R---	-R---
008D0000	00046000			MAP	-R---	-R---
00916000	013BA000	Reserved (008D0000)		MAP		-R---
6FBE0000	00001000	zloader.dll		IMG	-R---	ERWC-
6FBE1000	0003D000	".text"	Executable code	IMG	ER---	ERWC-
6FC1E000	00025000	".rdata"	Read-only initialized data	IMG	-R---	ERWC-
6FC43000	00F19000	".data"	Initialized data	IMG	-RW--	ERWC-
70B5C000	00001000	".rsrc"	Resources	IMG	-R---	ERWC-
70B5D000	00002000	".reloc"	Base relocations	IMG	-R---	ERWC-
74B50000	0005C000			IMG	-R---	ERWC-
75740000	00008000			IMG	-R---	ERWC-
75750000	0003F000			IMG	-R---	ERWC-
759C0000	00001000	cryptbase.dll		IMG	-R---	ERWC-
759C1000	00008000	".text"	Executable code	IMG	ER---	ERWC-
759C9000	00001000	".data"	Initialized data	IMG	-RW--	ERWC-
759CA000	00001000	".rsrc"	Resources	IMG	-R---	ERWC-
759CB000	00001000	".reloc"	Base relocations	IMG	-R---	ERWC-
759D0000	00001000	sspicli.dll		IMG	-R---	ERWC-
759D1000	0000F000	Reserved (759D0000)		IMG		ERWC-
759E0000	00016000	sspicli.dll		IMG	ER---	ERWC-
759F6000	0000A000	Reserved (759D0000)		IMG		ERWC-
75A00000	00001000	sspicli.dll		IMG	-RW--	ERWC-

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

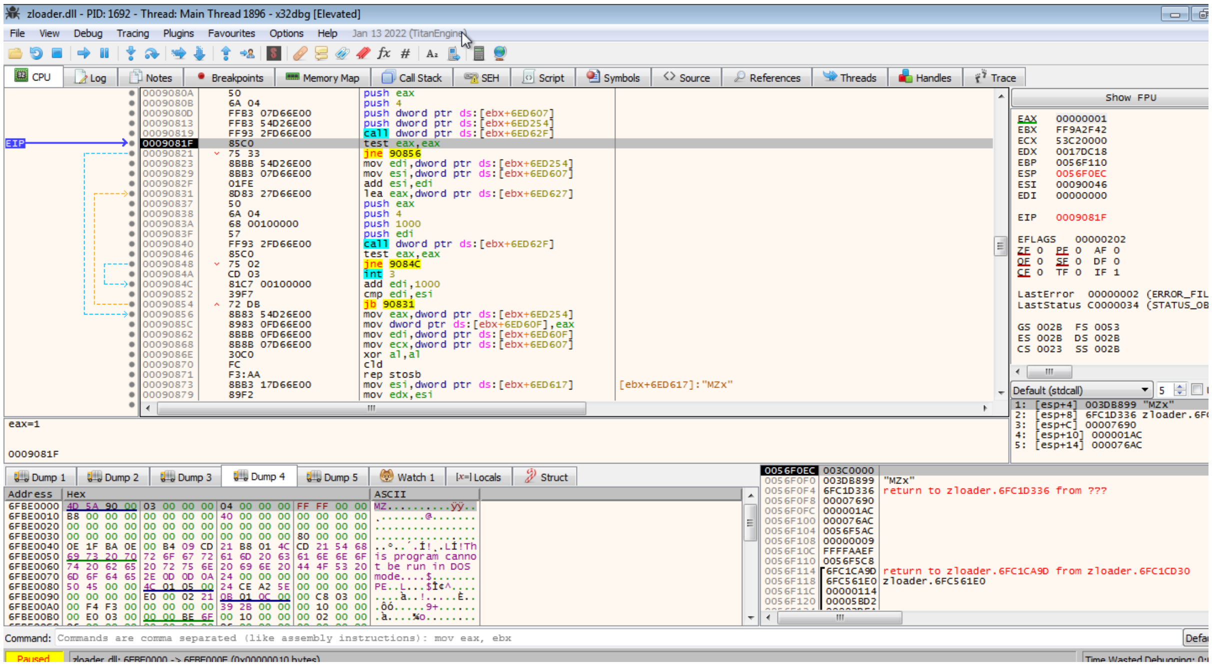
Paused zloader.dll: 6FBE0000 -> 6FBE000F (0x00000010 bytes) Time Wasted Debugging

After the call get executed that permission is changed to read,write.


zloader.dll - PID: 1692 - Module: kernelbase.dll - Thread: Main Thread 1896 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace

```
Address  Size     Info                                  Content                     Type Protection Initial
00010000 00010000                                                                   MAP  -RW--      -RW--
00020000 00010000                                                                   MAP  -RW--      -RW--
00030000 00001000                                                                   PRV  -RW--      -RW--
00040000 00001000                                                                   IMG  -R---      ERWC-
00050000 00004000                                                                   MAP  -R---      -R---
00060000 00001000                                                                   PRV  -RW--      -RW--
00070000 00001000                                                                   PRV  -RW--      -RW--
00080000 00001000                                                                   MAP  -R---      -R---
00090000 00001000                                                                   PRV  ERW--      ERW--
000C0000 00006000                                                                   PRV  -RW--      -RW--
000C6000 0007A000 Reserved (000C0000)                                               PRV             -RW--
00140000 00039000 Reserved                                                          PRV             -RW--
00179000 00007000                                                                   PRV  -RW-G      -RW--
001D0000 0000E000                                                                   PRV  -RW--      -RW--
001DE000 000F2000 Reserved (001D0000)                                               PRV             -RW--
002D0000 00067000 \Device\HarddiskVolume1\Windows\S                                 MAP  -R---      -R---
003A0000 00001000 dllloader32_bbb1.exe                                              IMG  -R---      ERWC-
003A1000 0000A000  ".text"                              Executable code             IMG  ER---      ERWC-
003AB000 00005000  ".rdata"                             Read-only initialized data  IMG  -R---      ERWC-
003B0000 00004000  ".data"                              Initialized data            IMG  -RW--      ERWC-
003B4000 00001000  ".reloc"                             Base relocations            IMG  -R---      ERWC-
003C0000 00082000                                                                   PRV  ERW--      ERW--
00470000 000FC000 Reserved                                                          PRV             -RW--
0056C000 00004000 Thread 768 Stack                                                  PRV  -RW-G      -RW--
00570000 00004000                                                                   MAP  -R---      -R---
00574000 0017C000 Reserved (00570000)                                               MAP             -R---
006F0000 00003000                                                                   MAP  -R---      -R---
006F3000 00005000 Reserved (00570000)                                               MAP             -R---
00730000 00003000                                                                   PRV  -RW--      -RW--
00733000 0000D000 Reserved (00730000)                                               PRV             -RW--
00740000 00181000                                                                   MAP  -R---      -R---
008D0000 00046000                                                                   MAP  -R---      -R---
00916000 013BA000 Reserved (008D0000)                                               MAP             -R---
6FBE0000 00001000 zloader.dll                                                       IMG  -RWC-      ERWC-
6FBE1000 0003D000  ".text"                              Executable code             IMG  -RWC-      ERWC-
6FC1E000 00025000  ".rdata"                             Read-only initialized data  IMG  -R---      ERWC-
6FC43000 00F19000  ".data"                              Initialized data            IMG  -RW--      ERWC-
70B5C000 00001000  ".rsrc"                              Resources                   IMG  -R---      ERWC-
70B5D000 00002000  ".reloc"                             Base relocations            IMG  -R---      ERWC-
74B50000 0005C000                                                                   IMG  -R---      ERWC-
75740000 00008000                                                                   IMG  -R---      ERWC-
75750000 0003F000                                                                   IMG  -R---      ERWC-
759C0000 00001000 cryptbase.dll                                                     IMG  -R---      ERWC-
759C1000 00008000  ".text"                              Executable code             IMG  ER---      ERWC-
759C9000 00001000  ".data"                              Initialized data            IMG  -RW--      ERWC-
759CA000 00001000  ".rsrc"                              Resources                   IMG  -R---      ERWC-
759CB000 00001000  ".reloc"                             Base relocations            IMG  -R---      ERWC-
759D0000 00001000 sspicli.dll                                                       IMG  -R---      ERWC-
759D1000 0000F000 Reserved (759D0000)                                               IMG             ERWC-
759E0000 00016000 sspicli.dll                                                       IMG  ER---      ERWC-
759F6000 0000A000 Reserved (759D0000)                                               IMG             ERWC-
75A00000 00001000 sspicli.dll                                                       IMG  -RW--      ERWC-
```

We are now return here i.e. to user code, till now the unpacking is completed ,now probably it will take jump to unpack code either by jump to eax or by some register in this case there is not any jump to register here it will use abnormal ret , just step over .


zloader.dll - PID: 1692 - Thread: Main Thread 1896 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trace
EIP
Show FPU
Dump 1 Dump 2 Dump 3 Dump 4 Dump 5 Watch 1 Locals Struct
Paused
Time Wasted Debugging: 0:

```
0009080A   50                push eax
0009080B   6A 04             push 4
0009080D   FFB3 07D66E00     push dword ptr ds:[ebx+6ED607]
00090813   FFB3 54D26E00     push dword ptr ds:[ebx+6ED254]
00090819   FF93 2FD66E00     call dword ptr ds:[ebx+6ED62F]
0009081F   85C0              test eax,eax
00090821   75 33             jne 90856
00090823   8BBB 54D26E00     mov edi,dword ptr ds:[ebx+6ED254]
00090829   8BB3 07D66E00     mov esi,dword ptr ds:[ebx+6ED607]
0009082F   01FE              add esi,edi
00090831   8D83 27D66E00     lea eax,dword ptr ds:[ebx+6ED627]
00090837   50                push eax
00090838   6A 04             push 4
0009083A   68 00100000       push 1000
0009083F   57                push edi
00090840   FF93 2FD66E00     call dword ptr ds:[ebx+6ED62F]
00090846   85C0              test eax,eax
00090848   75 02             jne 9084C
0009084A   CD 03             int 3
0009084C   81C7 00100000     add edi,1000
00090852   39F7              cmp edi,esi
00090854   72 DB             jb 90831
00090856   8B83 54D26E00     mov eax,dword ptr ds:[ebx+6ED254]
0009085C   8983 0FD66E00     mov dword ptr ds:[ebx+6ED60F],eax
00090862   8BBB 0FD66E00     mov edi,dword ptr ds:[ebx+6ED60F]
00090868   8B8B 07D66E00     mov ecx,dword ptr ds:[ebx+6ED607]
0009086E   30C0              xor al,al
00090870   FC                cld
00090871   F3:AA             rep stosb
00090873   8BB3 17D66E00     mov esi,dword ptr ds:[ebx+6ED617]     [ebx+6ED617]:"MZx"
00090879   89F2              mov edx,esi

eax=1

0009081F
```

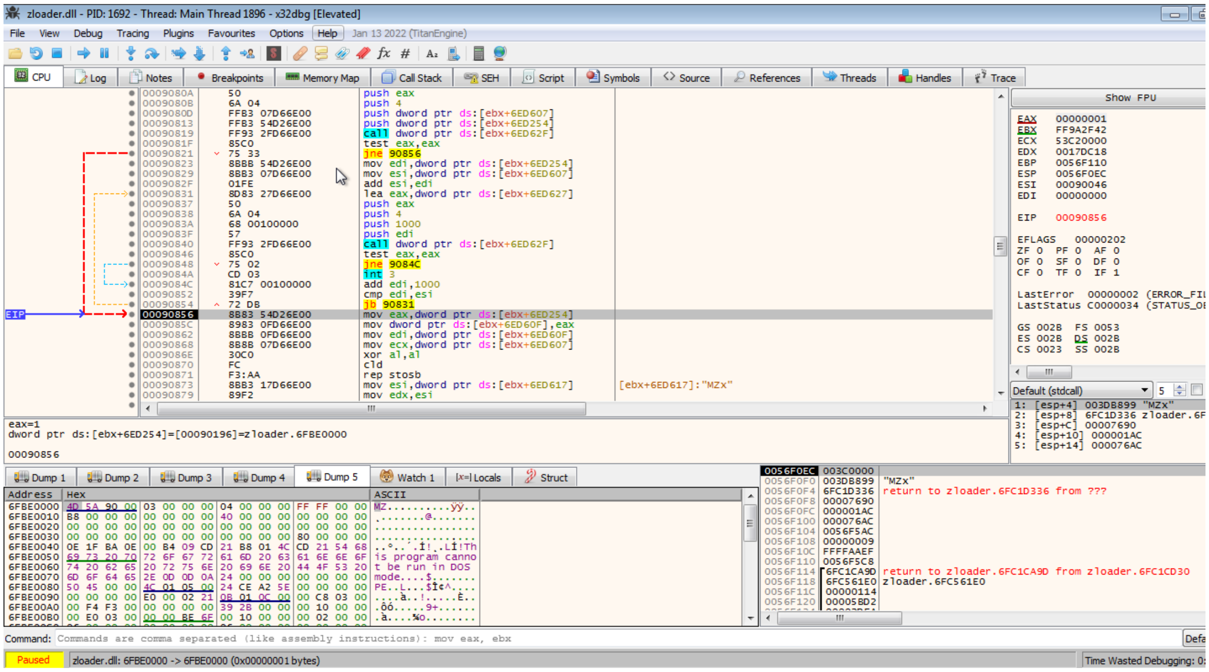
```
EAX    00000001
EBX    FF9A2F42
ECX    53C20000
EDX    0017DC18
EBP    0056F110
ESP    0056F0EC
ESI    00090046
EDI    00000000

EIP    0009081F

EFLAGS    00000202
ZF 0   PF 0   AF 0
OF 0   SF 0   DF 0
CF 0   TF 0   IF 1

LastError  00000002 (ERROR_FIL
LastStatus C0000034 (STATUS_OB

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

Default (stdcall)   5
1:  [esp+4] 003DB899 "MZx"
2:  [esp+8] 6FC1D336 zloader.6FC
3:  [esp+C] 00007690
4:  [esp+10] 000001AC
5:  [esp+14] 000076AC
```

```
Address  Hex                                              ASCII
6FBE0000 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
6FBE0010 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
6FBE0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
6FBE0030 00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00  ................
6FBE0040 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
6FBE0050 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
6FBE0060 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
6FBE0070 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
6FBE0080 50 45 00 00 4C 01 05 00 24 CE A2 5E 00 00 00 00  PE..L...$Î¢^....
6FBE0090 00 00 00 00 E0 00 02 21 0B 01 0C 00 00 C8 03 00  ....à..!.....È..
6FBE00A0 00 F4 F3 00 00 00 00 00 39 2B 00 00 00 10 00 00  .ôó.....9+......
6FBE00B0 00 E0 03 00 00 00 BE 6F 00 10 00 00 00 02 00 00  .à....¾o........
```

```
0056F0EC 003C0000
0056F0F0 003DB899  "MZx"
0056F0F4 6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8 00007690
0056F0FC 000001AC
0056F100 000076AC
0056F104 0056F5AC
0056F108 00000009
0056F10C FFFFAAEF
0056F110 0056F5C8
0056F114 6FC1CA9D  return to zloader.6FC1CA9D from zloader.6FC1CD30
0056F118 6FC561E0  zloader.6FC561E0
0056F11C 00000114
0056F120 00005BD2
```

```
Command: Commands are comma separated (like assembly instructions): mov eax, ebx
zloader.dll: 6FBE0000 -> 6FBE000F (0x00000010 bytes)
```

Now we need to check all these memory location in dump for executable.



```
zloader.dll - PID: 1692 - Thread: Main Thread 1896 - x32dbg [Elevated]
```

```
0009080A   50                push eax
0009080B   6A 04             push 4
0009080D   FFB3 07D66E00     push dword ptr ds:[ebx+6ED607]
00090813   FFB3 54D26E00     push dword ptr ds:[ebx+6ED254]
00090819   FF93 2FD66E00     call dword ptr ds:[ebx+6ED62F]
0009081F   85C0              test eax,eax
00090821   75 33             jne 90856
00090823   8BBB 54D26E00     mov edi,dword ptr ds:[ebx+6ED254]
00090829   8BB3 07D66E00     mov esi,dword ptr ds:[ebx+6ED607]
0009082F   01FE              add esi,edi
00090831   8D83 27D66E00     lea eax,dword ptr ds:[ebx+6ED627]
00090837   50                push eax
00090838   6A 04             push 4
0009083A   68 00100000       push 1000
0009083F   57                push edi
00090840   FF93 2FD66E00     call dword ptr ds:[ebx+6ED62F]
00090846   85C0              test eax,eax
00090848   75 02             jne 9084C
0009084A   CD 03             int 3
0009084C   81C7 00100000     add edi,1000
00090852   39F7              cmp edi,esi
00090854   72 DB             jb 90831
EIP  00090856   8B83 54D26E00     mov eax,dword ptr ds:[ebx+6ED254]
0009085C   8983 0FD66E00     mov dword ptr ds:[ebx+6ED60F],eax
00090862   8BBB 0FD66E00     mov edi,dword ptr ds:[ebx+6ED60F]
00090868   8B8B 07D66E00     mov ecx,dword ptr ds:[ebx+6ED607]
0009086E   30C0              xor al,al
00090870   FC                cld
00090871   F3:AA             rep stosb
00090873   8BB3 17D66E00     mov esi,dword ptr ds:[ebx+6ED617]     [ebx+6ED617]:"MZx"
00090879   89F2              mov edx,esi

eax=1
dword ptr ds:[ebx+6ED254]=[00090196]=zloader.6FBE0000

00090856
```

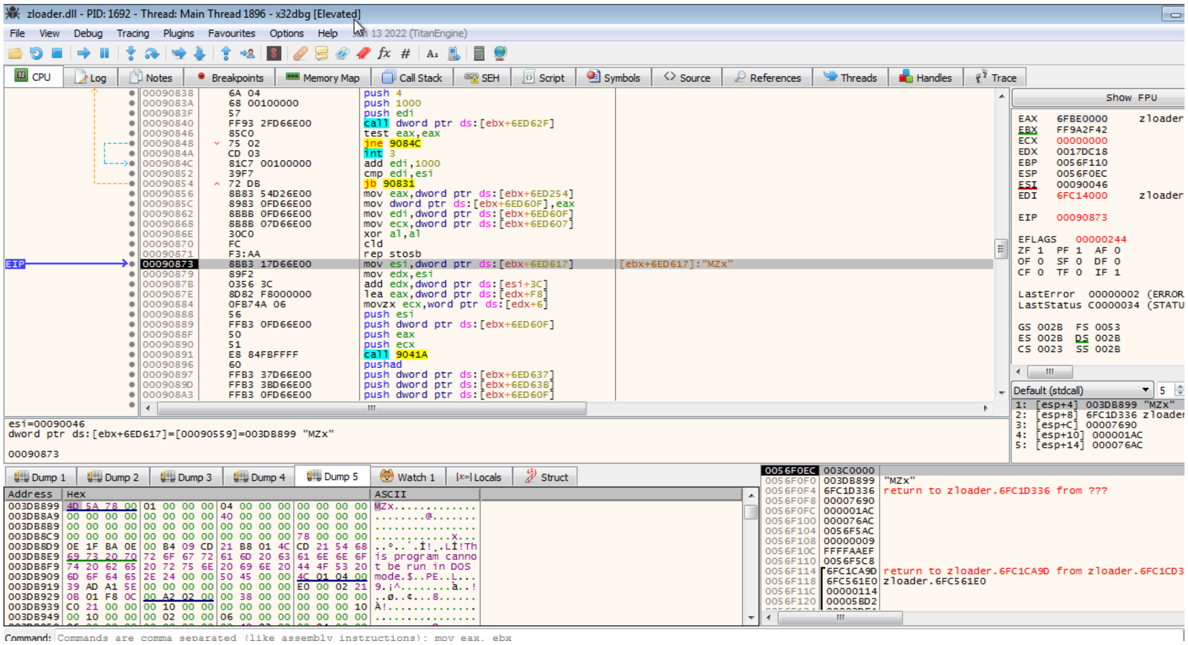
```
EAX     00000001
EBX     FF9A2F42
ECX     53C20000
EDX     0017DC18
EBP     0056F110
ESP     0056F0EC
ESI     00090046
EDI     00000000

EIP     00090856

EFLAGS    00000202
ZF 0  PF 0  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError   00000002 (ERROR_FI
LastStatus  C0000034 (STATUS_OE

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

Default (stdcall)  5
1: [esp+4] 003DB899 "MZx"
2: [esp+8] 6FC1D336 zloader.6F
3: [esp+C] 00007690
4: [esp+10] 000001AC
5: [esp+14] 000076AC
```

```
Address   Hex                                              ASCII
6FBE0000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
6FBE0010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
6FBE0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
6FBE0030  00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00  ................
6FBE0040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
6FBE0050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
6FBE0060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
6FBE0070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
6FBE0080  50 45 00 00 4C 01 05 00 24 CE A2 5E 00 00 00 00  PE..L...$Î¢^....
6FBE0090  00 00 00 00 E0 00 02 21 0B 01 0C 00 00 C8 03 00  ....à..!.....È..
6FBE00A0  00 F4 F3 00 00 00 00 00 39 2B 00 00 00 10 00 00  .ôó.....9+......
6FBE00B0  00 E0 03 00 00 00 BE 6F 00 10 00 00 00 02 00 00  .à....¾o........
```

```
0056F0EC  003C0000
0056F0F0  003DB899  "MZx"
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8  00007690
0056F0FC  000001AC
0056F100  000076AC
0056F104  0056F5AC
0056F108  00000009
0056F10C  FFFFAAEF
0056F110  0056F5C8
0056F114  6FC1CA9D  return to zloader.6FC1CA9D from zloader.6FC1CD30
0056F118  6FC561E0  zloader.6FC561E0
0056F11C  00000114
0056F120  00005BD2
```

```
Command: Commands are comma separated (like assembly instructions): mov eax, ebx
Paused  zloader.dll: 6FBE0000 -> 6FBE0000 (0x00000001 bytes)    Time Wasted Debugging: 0:
```

finally we got the location which is containing the executable and we also dump that in dump5.



```
zloader.dll - PID: 1692 - Thread: Main Thread 1896 - x32dbg [Elevated]
```

```
00090838   6A 04             push 4
0009083A   68 00100000       push 1000
0009083F   57                push edi
00090840   FF93 2FD66E00     call dword ptr ds:[ebx+6ED62F]
00090846   85C0              test eax,eax
00090848   75 02             jne 9084C
0009084A   CD 03             int 3
0009084C   81C7 00100000     add edi,1000
00090852   39F7              cmp edi,esi
00090854   72 DB             jb 90831
00090856   8B83 54D26E00     mov eax,dword ptr ds:[ebx+6ED254]
0009085C   8983 0FD66E00     mov dword ptr ds:[ebx+6ED60F],eax
00090862   8BBB 0FD66E00     mov edi,dword ptr ds:[ebx+6ED60F]
00090868   8B8B 07D66E00     mov ecx,dword ptr ds:[ebx+6ED607]
0009086E   30C0              xor al,al
00090870   FC                cld
00090871   F3:AA             rep stosb
EIP  00090873   8BB3 17D66E00     mov esi,dword ptr ds:[ebx+6ED617]     [ebx+6ED617]:"MZx"
00090879   89F2              mov edx,esi
0009087B   0356 3C           add edx,dword ptr ds:[esi+3C]
0009087E   8D82 F8000000     lea eax,dword ptr ds:[edx+F8]
00090884   0FB74A 06         movzx ecx,word ptr ds:[edx+6]
00090888   56                push esi
00090889   FFB3 0FD66E00     push dword ptr ds:[ebx+6ED60F]
0009088F   50                push eax
00090890   51                push ecx
00090891   E8 84FBFFFF       call 9041A
00090896   60                pushad
00090897   FFB3 37D66E00     push dword ptr ds:[ebx+6ED637]
0009089D   FFB3 3BD66E00     push dword ptr ds:[ebx+6ED63B]
000908A3   FFB3 0FD66E00     push dword ptr ds:[ebx+6ED60F]

esi=00090046
dword ptr ds:[ebx+6ED617]=[00090559]=003DB899 "MZx"

00090873
```

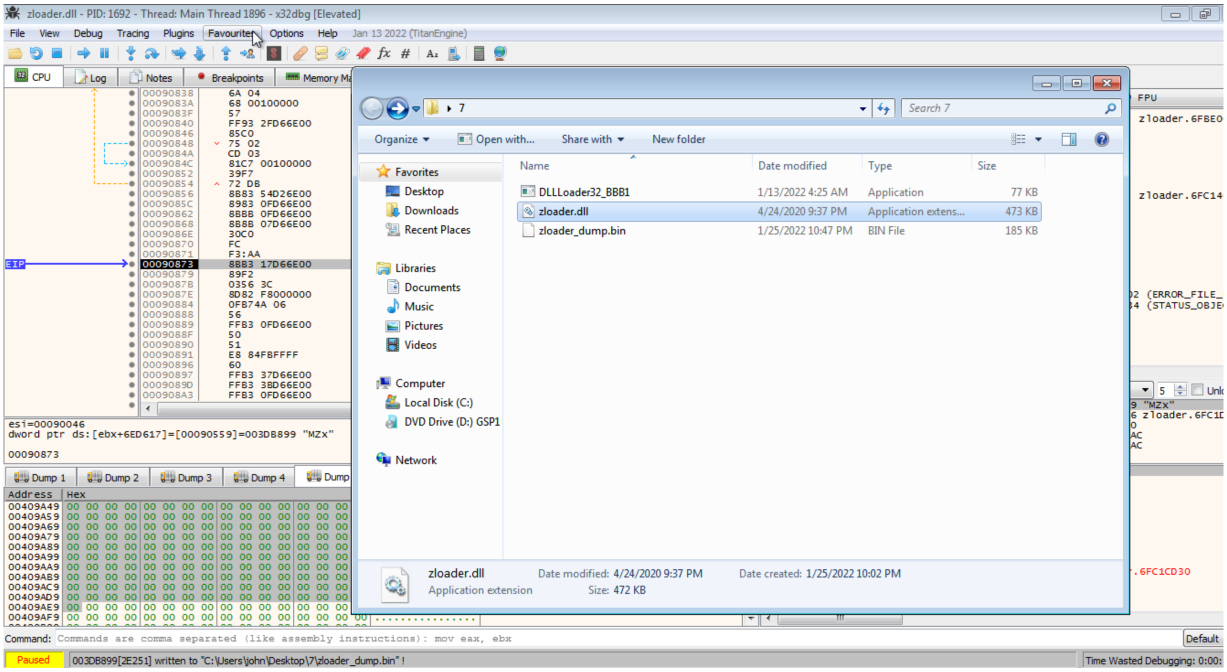
```
EAX     6FBE0000     zloader
EBX     FF9A2F42
ECX     00000000
EDX     0017DC18
EBP     0056F110
ESP     0056F0EC
ESI     00090046
EDI     6FC14000     zloader

EIP     00090873

EFLAGS    00000244
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError   00000002 (ERROR
LastStatus  C0000034 (STATU

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

Default (stdcall)  5
1: [esp+4] 003DB899 "MZx"
2: [esp+8] 6FC1D336 zloader
3: [esp+C] 00007690
4: [esp+10] 000001AC
5: [esp+14] 000076AC
```

```
Address   Hex                                              ASCII
003DB899  4D 5A 78 00 01 00 00 00 04 00 00 00 00 00 00 00  MZx.............
003DB8A9  00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ........@.......
003DB8B9  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
003DB8C9  00 00 00 00 00 00 00 00 00 00 00 00 78 00 00 00  ............x...
003DB8D9  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
003DB8E9  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
003DB8F9  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
003DB909  6D 6F 64 65 2E 24 00 00 50 45 00 00 4C 01 04 00  mode.$..PE..L...
003DB919  39 AD A1 5E 00 00 00 00 00 00 00 00 E0 00 02 21  9.¡^........à..!
003DB929  0B 01 F8 0C 00 A2 02 00 00 38 00 00 00 00 00 00  ..ø..¢...8......
003DB939  C0 21 00 00 00 10 00 00 00 00 00 00 00 00 00 10  À!..............
003DB949  00 10 00 00 00 02 00 00 06 00 00 00 00 00 00 00  ................
```

```
0056F0EC  003C0000
0056F0F0  003DB899  "MZx"
0056F0F4  6FC1D336  return to zloader.6FC1D336 from ???
0056F0F8  00007690
0056F0FC  000001AC
0056F100  000076AC
0056F104  0056F5AC
0056F108  00000009
0056F10C  FFFFAAEF
0056F110  0056F5C8
0056F114  6FC1CA9D  return to zloader.6FC1CA9D from zloader.6FC1CD3
0056F118  6FC561E0  zloader.6FC561E0
0056F11C  00000114
0056F120  00005BD2
```

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

Now we dump the file here.


zloader.dll - PID: 1692 - Thread: Main Thread 1896 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Jan 13 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Ma
Dump 1 Dump 2 Dump 3 Dump 4 Dump
Organize Open with... Share with New folder
Search 7
Favorites Desktop Downloads Recent Places
Libraries Documents Music Pictures Videos
Computer Local Disk (C:) DVD Drive (D:) GSP1
Network

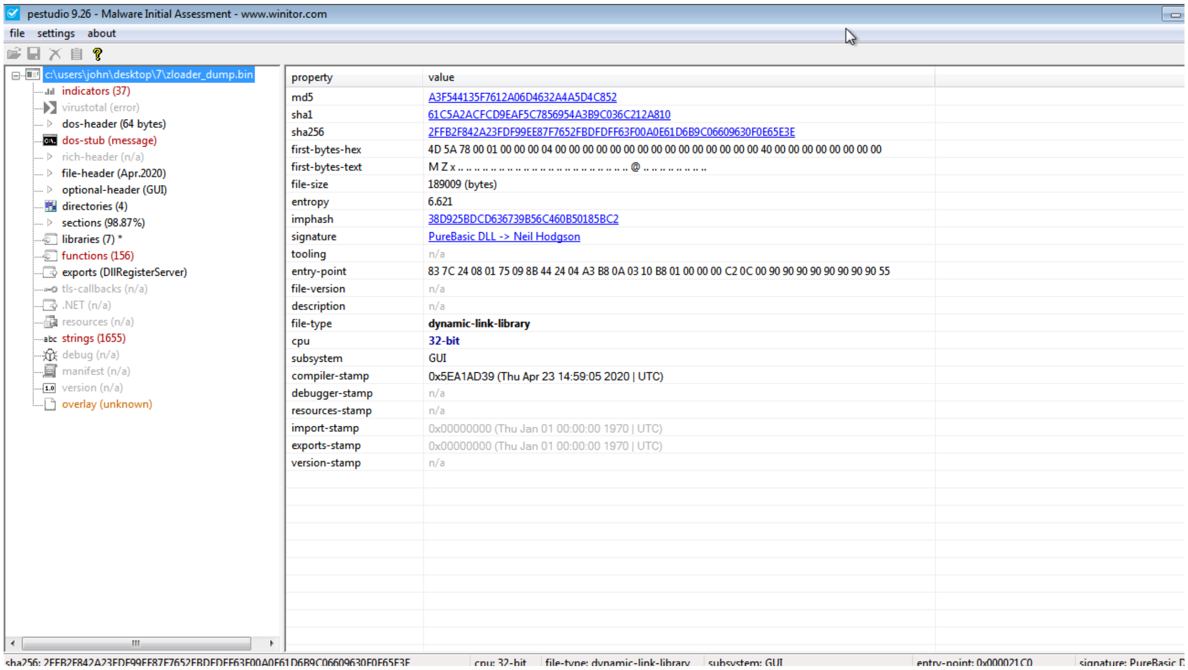
```
00090838   6A 04
0009083A   68 00100000
0009083F   57
00090840   FF93 2FD66E00
00090846   85C0
00090848   75 02
0009084A   CD 03
0009084C   81C7 00100000
00090852   39F7
00090854   72 DB
00090856   8B83 54D26E00
0009085C   8983 0FD66E00
00090862   8BBB 0FD66E00
00090868   8B8B 07D66E00
0009086E   30C0
00090870   FC
00090871   F3:AA
EIP 00090873   8BB3 17D66E00
00090879   89F2
0009087B   0356 3C
0009087E   8D82 F8000000
00090884   0FB74A 06
00090888   56
00090889   FFB3 0FD66E00
0009088F   50
00090890   51
00090891   E8 84FBFFFF
00090896   60
00090897   FFB3 37D66E00
0009089D   FFB3 3BD66E00
000908A3   FFB3 0FD66E00

esi=00090046
dword ptr ds:[ebx+6ED617]=[00090559]=003DB899 "MZx"

00090873
```

Name	Date modified	Type	Size
DLLLoader32_BBB1	1/13/2022 4:25 AM	Application	77 KB
zloader.dll	4/24/2020 9:37 PM	Application extens...	473 KB
zloader_dump.bin	1/25/2022 10:47 PM	BIN File	185 KB

zloader.dll Date modified: 4/24/2020 9:37 PM Date created: 1/25/2022 10:02 PM
Application extension Size: 472 KB

```
Command: Commands are comma separated (like assembly instructions): mov eax, ebx
Paused  003DB899[2E251] written to "C:\Users\john\Desktop\7\zloader_dump.bin" !   Time Wasted Debugging: 0:00:
```

Now here we load the dumped file in pestudio,here the entropy is low which means that we have successfully unpack it.

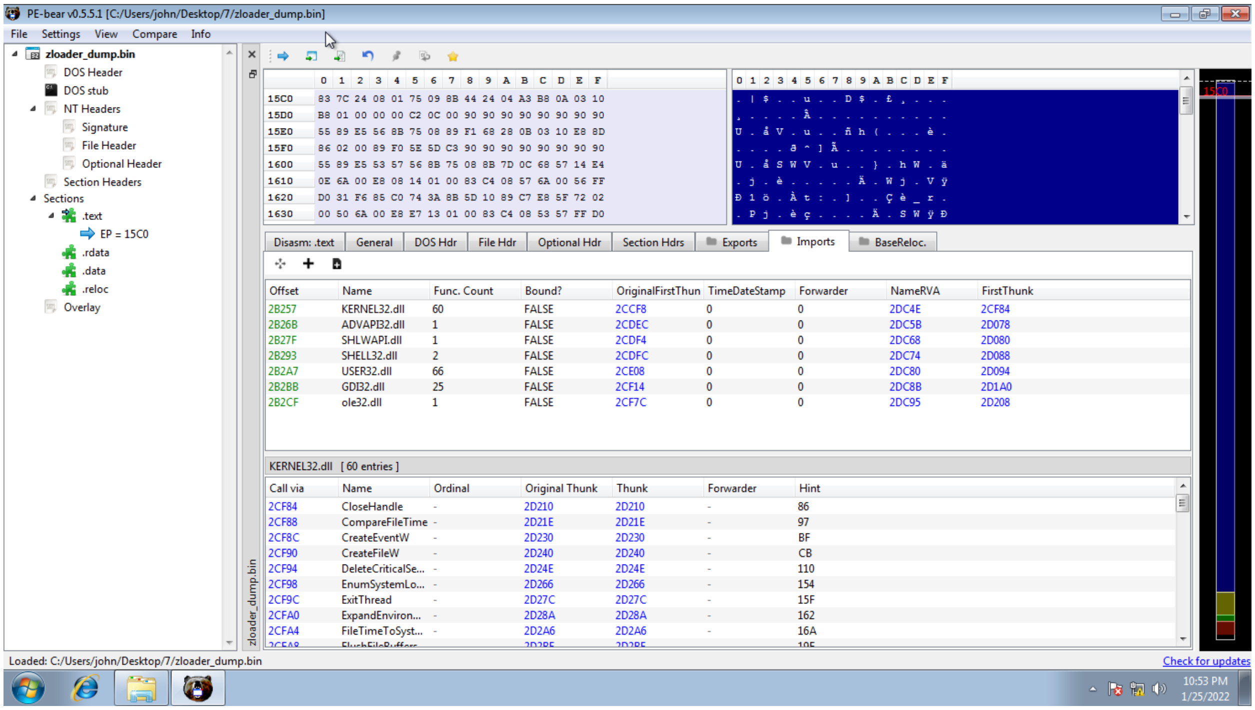

pestudio 9.26 - Malware Initial Assessment - www.winitor.com
file settings about

c:\users\john\desktop\7\zloader_dump.bin
- indicators (37)
- virustotal (error)
- dos-header (64 bytes)
- dos-stub (message)
- rich-header (n/a)
- file-header (Apr.2020)
- optional-header (GUI)
- directories (4)
- sections (98.87%)
- libraries (7) *
- functions (156)
- exports (DllRegisterServer)
- tls-callbacks (n/a)
- .NET (n/a)
- resources (n/a)
- strings (1655)
- debug (n/a)
- manifest (n/a)
- version (n/a)
- overlay (unknown)

property	value
md5	A3F544135F7612A06D4632A4A5D4C852
sha1	61C5A2ACFCD9EAF5C7856954A3B9C036C212A810
sha256	2FFB2F842A23FDF99EE87F7652FBDFDFF63F00A0E61D6B9C06609630F0E65E3E
first-bytes-hex	4D 5A 78 00 01 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00
first-bytes-text	M Z x .. .. @ .. ..
file-size	189009 (bytes)
entropy	6.621
imphash	38D925BDCD636739B56C460B50185BC2
signature	PureBasic DLL -> Neil Hodgson
tooling	n/a
entry-point	83 7C 24 08 01 75 09 8B 44 24 04 A3 B8 0A 03 10 B8 01 00 00 00 C2 0C 00 90 90 90 90 90 90 90 90 55
file-version	n/a
description	n/a
file-type	**dynamic-link-library**
cpu	**32-bit**
subsystem	GUI
compiler-stamp	0x5EA1AD39 (Thu Apr 23 14:59:05 2020 | UTC)
debugger-stamp	n/a
resources-stamp	n/a
import-stamp	0x00000000 (Thu Jan 01 00:00:00 1970 | UTC)
exports-stamp	0x00000000 (Thu Jan 01 00:00:00 1970 | UTC)
version-stamp	n/a

sha256: 2FFB2F842A23FDF99EE87F7652FBDFDFF63F00A0E61D6B9C06609630F0E65E3E | cpu: 32-bit | file-type: dynamic-link-library | subsystem: GUI | entry-point: 0x000021C0 | signature: PureBasic D

Here now we load the sample in PE-bear and we can see the imports.



```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F     0 1 2 3 4 5 6 7 8 9 A B C D E F
15C0  83 7C 24 08 01 75 09 8B 44 24 04 A3 B8 0A 03 10    . | $ . . u . . D $ . £ ¸ . . .
15D0  B8 01 00 00 00 C2 0C 00 90 90 90 90 90 90 90 90    ¸ . . . . Â . . . . . . . . . .
15E0  55 89 E5 56 8B 75 08 89 F1 68 28 0B 03 10 E8 8D    U . å V . u . . ñ h ( . . . è .
15F0  86 02 00 89 F0 5E 5D C3 90 90 90 90 90 90 90 90    . . . . ð ^ ] Ã . . . . . . . .
1600  55 89 E5 53 57 56 8B 75 08 8B 7D 0C 68 57 14 E4    U . å S W V . u . . } . h W . ä
1610  0E 6A 00 E8 08 14 01 00 83 C4 08 57 6A 00 56 FF    . j . è . . . . . Ä . W j . V ÿ
1620  D0 31 F6 85 C0 74 3A 8B 5D 10 89 C7 E8 5F 72 02    Ð 1 ö . À t : . ] . . Ç è _ r .
1630  00 50 6A 00 E8 E7 13 01 00 83 C4 08 53 57 FF D0    . P j . è ç . . . . Ä . S W ÿ Ð
```

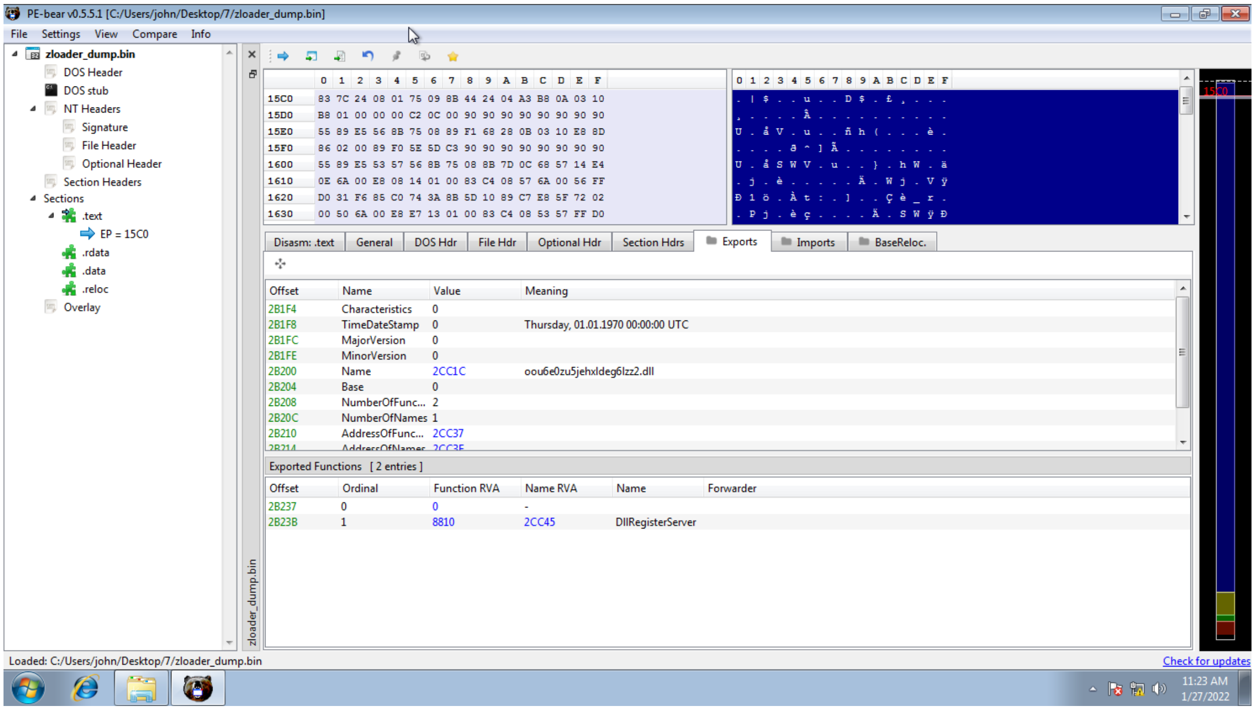
Offset	Name	Func. Count	Bound?	OriginalFirstThun	TimeDateStamp	Forwarder	NameRVA	FirstThunk
2B257	KERNEL32.dll	60	FALSE	2CCF8	0	0	2DC4E	2CF84
2B26B	ADVAPI32.dll	1	FALSE	2CDEC	0	0	2DC5B	2D078
2B27F	SHLWAPI.dll	1	FALSE	2CDF4	0	0	2DC68	2D080
2B293	SHELL32.dll	2	FALSE	2CDFC	0	0	2DC74	2D088
2B2A7	USER32.dll	66	FALSE	2CE08	0	0	2DC80	2D094
2B2BB	GDI32.dll	25	FALSE	2CF14	0	0	2DC8B	2D1A0
2B2CF	ole32.dll	1	FALSE	2CF7C	0	0	2DC95	2D208

KERNEL32.dll [60 entries]

Call via	Name	Ordinal	Original Thunk	Thunk	Forwarder	Hint
2CF84	CloseHandle	-	2D210	2D210	-	86
2CF88	CompareFileTime	-	2D21E	2D21E	-	97
2CF8C	CreateEventW	-	2D230	2D230	-	BF
2CF90	CreateFileW	-	2D240	2D240	-	CB
2CF94	DeleteCriticalSe...	-	2D24E	2D24E	-	110
2CF98	EnumSystemLo...	-	2D266	2D266	-	154
2CF9C	ExitThread	-	2D27C	2D27C	-	15F
2CFA0	ExpandEnviron...	-	2D28A	2D28A	-	162
2CFA4	FileTimeToSyst...	-	2D2A6	2D2A6	-	16A

Loaded: C:/Users/john/Desktop/7/zloader_dump.bin

Here we can see the exports, when this will run it will be using DllRegisterServer.



```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F     0 1 2 3 4 5 6 7 8 9 A B C D E F
15C0  83 7C 24 08 01 75 09 8B 44 24 04 A3 B8 0A 03 10    . | $ . . u . . D $ . £ ¸ . . .
15D0  B8 01 00 00 00 C2 0C 00 90 90 90 90 90 90 90 90    ¸ . . . . Â . . . . . . . . . .
15E0  55 89 E5 56 8B 75 08 89 F1 68 28 0B 03 10 E8 8D    U . å V . u . . ñ h ( . . . è .
15F0  86 02 00 89 F0 5E 5D C3 90 90 90 90 90 90 90 90    . . . . ð ^ ] Ã . . . . . . . .
1600  55 89 E5 53 57 56 8B 75 08 8B 7D 0C 68 57 14 E4    U . å S W V . u . . } . h W . ä
1610  0E 6A 00 E8 08 14 01 00 83 C4 08 57 6A 00 56 FF    . j . è . . . . . Ä . W j . V ÿ
1620  D0 31 F6 85 C0 74 3A 8B 5D 10 89 C7 E8 5F 72 02    Ð 1 ö . À t : . ] . . Ç è _ r .
1630  00 50 6A 00 E8 E7 13 01 00 83 C4 08 53 57 FF D0    . P j . è ç . . . . Ä . S W ÿ Ð
```

Offset	Name	Value	Meaning
2B1F4	Characteristics	0	
2B1F8	TimeDateStamp	0	Thursday, 01.01.1970 00:00:00 UTC
2B1FC	MajorVersion	0	
2B1FE	MinorVersion	0	
2B200	Name	2CC1C	oou6e0zu5jehxldeg6lzz2.dll
2B204	Base	0	
2B208	NumberOfFunc...	2	
2B20C	NumberOfNames	1	
2B210	AddressOfFunc...	2CC37	

Exported Functions [2 entries]

Offset	Ordinal	Function RVA	Name RVA	Name	Forwarder
2B237	0	0	-		
2B23B	1	8810	2CC45	DllRegisterServer	

Loaded: C:/Users/john/Desktop/7/zloader_dump.bin