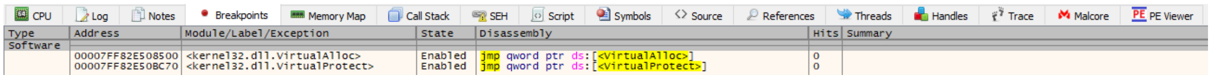
Reverse Engineering XOR Encryption

In this section I will be reverse engineering the xor shellcode in xdbg.

So, open the .exe file in xdbg, then here we will put some breakpoints, we will be 2 breakpoints:

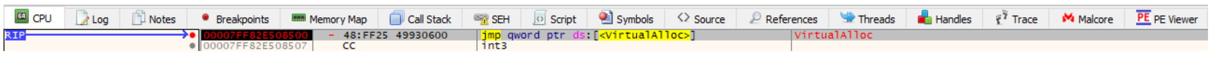
1. VirtualAlloc
2. VirtualProtect

Go to “Breakpoints” section and put these breakpoints.



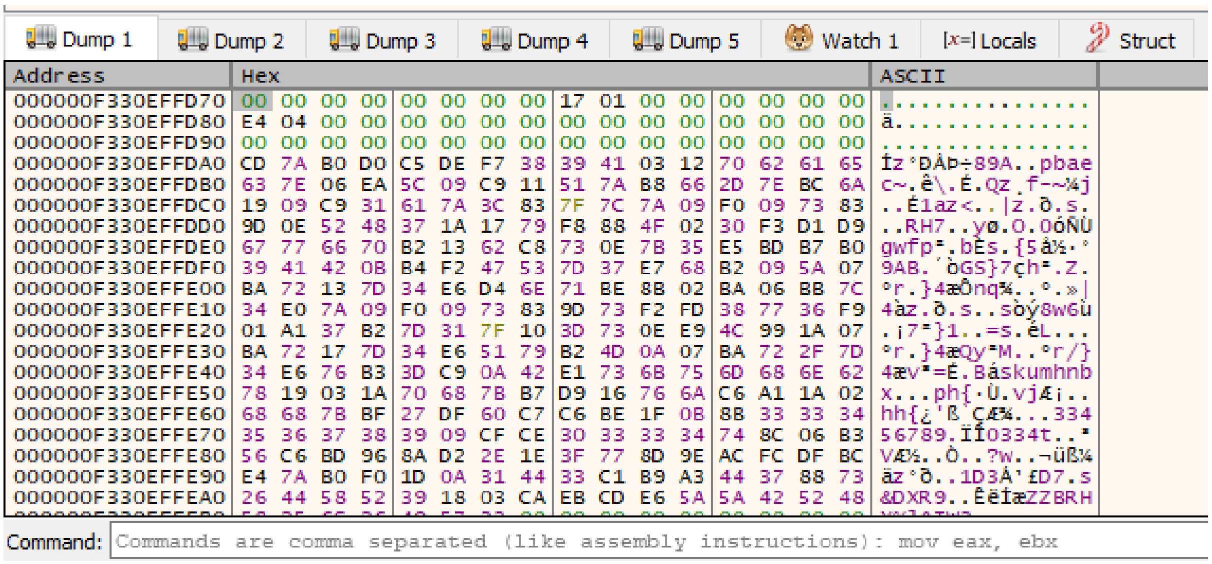
```
Type      Address           Module/Label/Exception            State    Disassembly                          Hits Summary
Software
          00007FF82E508500  <kernel32.dll.VirtualAlloc>       Enabled  jmp qword ptr ds:[<VirtualAlloc>]    0
          00007FF82E50BC70  <kernel32.dll.VirtualProtect>     Enabled  jmp qword ptr ds:[<VirtualProtect>]  0
```

Now run the program, we can see that we go to the first breakpoint, here we must follow the same steps as we followed in base64 part.



```
RIP  00007FF82E508500  - 48:FF25 49930600   jmp qword ptr ds:[<VirtualAlloc>]   VirtualAlloc
     00007FF82E508507    CC                 int3
```

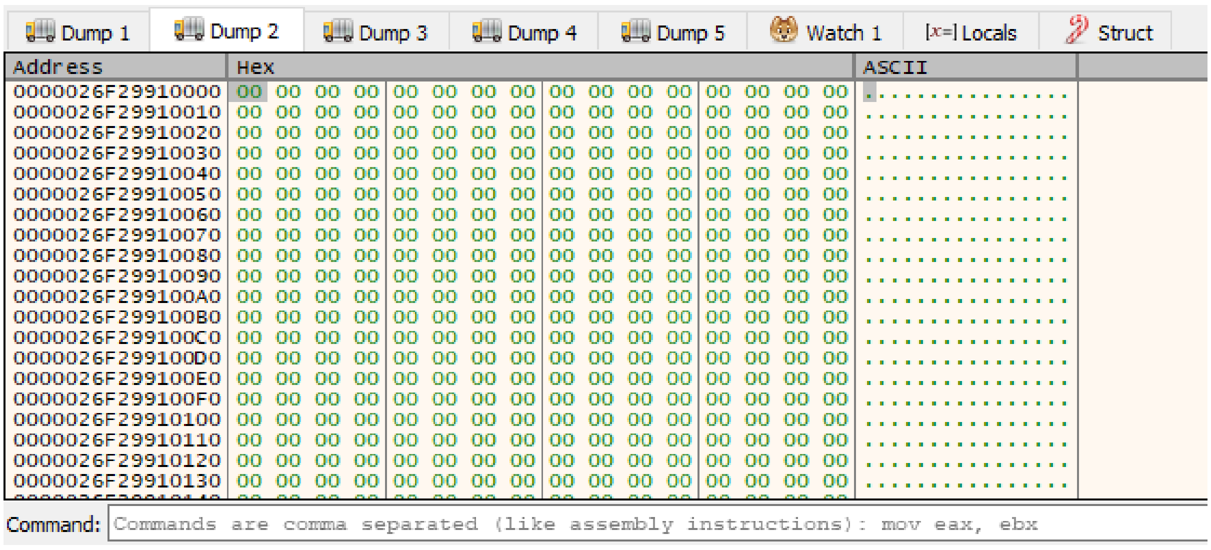
Then, step down, and go to the call VirtualAlloc, we know in the second parameter, we will get the address of the encrypted payload, so follow it in the dump1.



```
Address           Hex                                                ASCII
000000F330EFFD70  00 00 00 00 00 00 00 00 17 01 00 00 00 00 00 00  ................
000000F330EFFD80  E4 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ä...............
000000F330EFFD90  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F330EFFDA0  CD 7A B0 D0 C5 DE F7 38 39 41 03 12 70 62 61 65  Íz°ÐÅÞ÷89A..pbae
000000F330EFFDB0  63 7E 06 EA 5C 09 C9 11 51 7A B8 66 2D 7E BC 6A  c~.ê\.É.Qz¸f-~¼j
000000F330EFFDC0  19 09 C9 31 61 7A 3C 83 7F 7C 7A 09 F0 09 73 83  ..É1az<..|z.ð.s.
000000F330EFFDD0  9D 0E 52 48 37 1A 17 79 F8 88 4F 02 30 F3 D1 D9  ..RH7..yø.O.0óÑÙ
000000F330EFFDE0  67 77 66 70 B2 13 62 C8 73 0E 7B 35 E5 BD B7 B0  gwfp².bÈs.{5å½·°
000000F330EFFDF0  39 41 42 0B B4 F2 47 53 7D 37 E7 68 B2 09 5A 07  9AB.´òGS}7çh².Z.
000000F330EFFE00  BA 72 13 7D 34 E6 D4 6E 71 BE 8B 02 BA 06 BB 7C  ºr.}4æÔnq¾..º.»|
000000F330EFFE10  34 E0 7A 09 F0 09 73 83 9D 73 F2 FD 38 77 36 F9  4àz.ð.s..sòý8w6ù
000000F330EFFE20  01 A1 37 B2 7D 31 7F 10 3D 73 0E E9 4C 99 1A 07  .¡7²}1..=s.éL...
000000F330EFFE30  BA 72 17 7D 34 E6 51 79 B2 4D 0A 07 BA 72 2F 7D  ºr.}4æQy²M..ºr/}
000000F330EFFE40  34 E6 76 B3 3D C9 0A 42 E1 73 6B 75 6D 68 6E 62  4æv³=É.Báskumhnb
000000F330EFFE50  78 19 03 1A 70 68 7B B7 D9 16 76 6A C6 A1 1A 02  x...ph{·Ù.vjÆ¡..
000000F330EFFE60  68 68 7B BF 27 DF 60 C7 C6 BE 1F 0B 8B 33 33 34  hh{¿'ß`ÇÆ¾...334
000000F330EFFE70  35 36 37 38 39 09 CF CE 30 33 33 34 74 8C 06 B3  56789.ÏÎ0334t..³
000000F330EFFE80  56 C6 BD 96 8A D2 2E 1E 3F 77 8D 9E AC FC DF BC  VÆ½..Ò..?w..¬üß¼
000000F330EFFE90  E4 7A B0 F0 1D 0A 31 44 33 C1 B9 A3 44 37 88 73  äz°ð..1D3Á¹£D7.s
000000F330EFFEA0  26 44 58 52 39 18 03 CA EB CD E6 5A 5A 42 52 48  &DXR9..ÊëÍæZZBRH
```

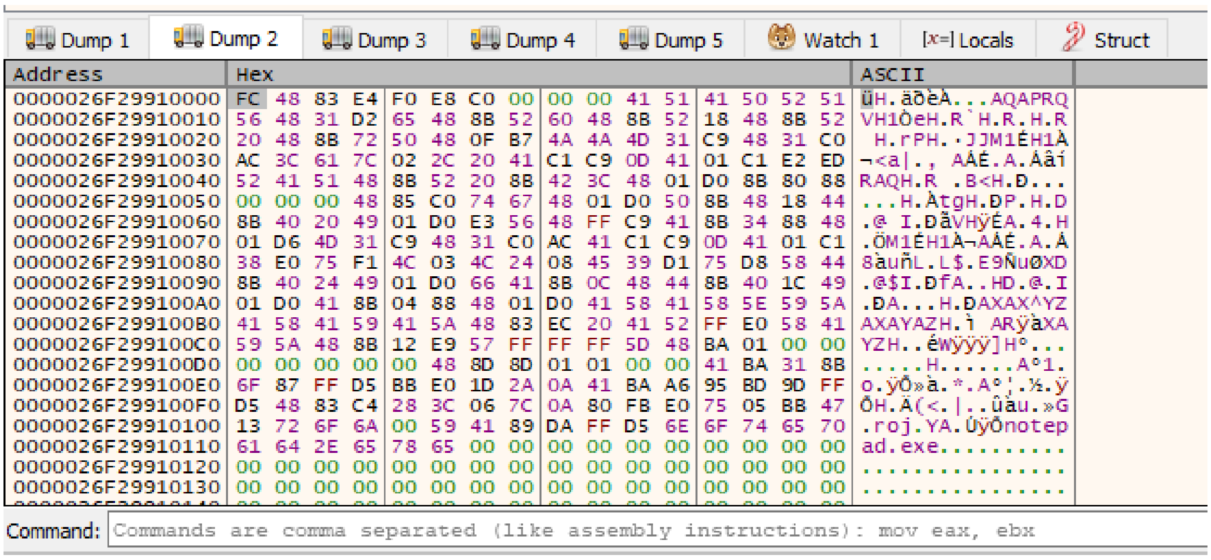
Command: Commands are comma separated (like assembly instructions): mov eax, ebx

Then once again step over, we can see the address here, so just copy it and follow it in the dump2, we can see that the space has been allocated for the decrypted payload.



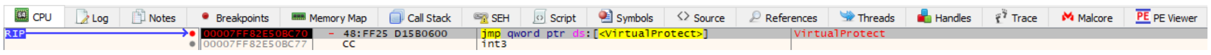
```
Address           Hex                                              ASCII
0000026F29910000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F299100A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F299100B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F299100C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F299100D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F299100E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F299100F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Command: Commands are comma separated (like assembly instructions): mov eax, ebx
```

So just run the program, and press enter in the cmd, we can see that in dump2 we get the payload.



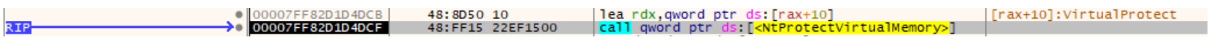
```
Address           Hex                                              ASCII
0000026F29910000  FC 48 83 E4 F0 E8 C0 00 00 00 41 51 41 50 52 51  üH.äðèÀ...AQAPRQ
0000026F29910010  56 48 31 D2 65 48 8B 52 60 48 8B 52 18 48 8B 52  VH1ÒeH.R`H.R.H.R
0000026F29910020  20 48 8B 72 50 48 0F B7 4A 4A 4D 31 C9 48 31 C0   H.rPH.·JJM1ÉH1À
0000026F29910030  AC 3C 61 7C 02 2C 20 41 C1 C9 0D 41 01 C1 E2 ED  ¬<a|., AÁÉ.A.Áâí
0000026F29910040  52 41 51 48 8B 52 20 8B 42 3C 48 01 D0 8B 80 88  RAQH.R .B<H.Ð...
0000026F29910050  00 00 00 48 85 C0 74 67 48 01 D0 50 8B 48 18 44  ...H.ÀtgH.ÐP.H.D
0000026F29910060  8B 40 20 49 01 D0 E3 56 48 FF C9 41 8B 34 88 48  .@ I.ÐãVHÿÉA.4.H
0000026F29910070  01 D6 4D 31 C9 48 31 C0 AC 41 C1 C9 0D 41 01 C1  .ÖM1ÉH1À¬AÁÉ.A.Á
0000026F29910080  38 E0 75 F1 4C 03 4C 24 08 45 39 D1 75 D8 58 44  8àuñL.L$.E9ÑuØXD
0000026F29910090  8B 40 24 49 01 D0 66 41 8B 0C 48 44 8B 40 1C 49  .@$I.ÐfA..HD.@.I
0000026F299100A0  01 D0 41 8B 04 88 48 01 D0 41 58 41 58 5E 59 5A  .ÐA...H.ÐAXAX^YZ
0000026F299100B0  41 58 41 59 41 5A 48 83 EC 20 41 52 FF E0 58 41  AXAYAZH.ì ARÿàXA
0000026F299100C0  59 5A 48 8B 12 E9 57 FF FF FF 5D 48 BA 01 00 00  YZH..éWÿÿÿ]Hº...
0000026F299100D0  00 00 00 00 00 48 8D 8D 01 01 00 00 41 BA 31 8B  .....H......Aº1.
0000026F299100E0  6F 87 FF D5 BB E0 1D 2A 0A 41 BA A6 95 BD 9D FF  o.ÿÕ»à.*.Aº¦.½.ÿ
0000026F299100F0  D5 48 83 C4 28 3C 06 7C 0A 80 FB E0 75 05 BB 47  ÕH.Ä(<.|..ûàu.»G
0000026F29910100  13 72 6F 6A 00 59 41 89 DA FF D5 6E 6F 74 65 70  .roj.YA.ÚÿÕnotep
0000026F29910110  61 64 2E 65 78 65 00 00 00 00 00 00 00 00 00 00  ad.exe..........
0000026F29910120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000026F29910130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
Command: Commands are comma separated (like assembly instructions): mov eax, ebx
```

We can even extract the shellcode from the VirtualProtect part.

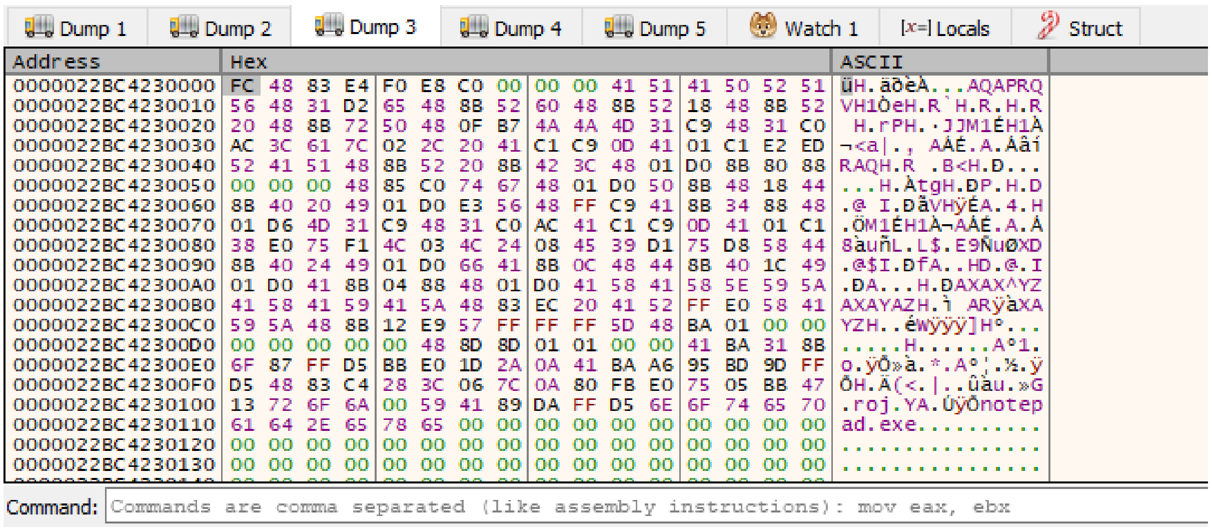


```
RIP ──────────> ● 00007FF82E50BC70  - 48:FF25 D15B0600  jmp qword ptr ds:[<VirtualProtect>]   VirtualProtect
                ● 00007FF82E50BC77    CC                int3
```

If we run the program, we will get a breakpoint at VirtualProtect, so step down to call function VirtualProtect, then check the second parameter, then follow it in dump3, we can see that we the shellcode, copy it and extract it, and compare it with the original shellcode, in Hexeditor, we can see that both payloads are same.



```
                ● 00007FF82D1D4DCB  48:8D50 10         lea rdx,qword ptr ds:[rax+10]                 [rax+10]:VirtualProtect
RIP ──────────> ● 00007FF82D1D4DCF  48:FF15 22EF1500   call qword ptr ds:[<NtProtectVirtualMemory>]
```



```
Address           Hex                                              ASCII
0000022BC4230000  FC 48 83 E4 F0 E8 C0 00 00 00 41 51 41 50 52 51  üH.äðèÀ...AQAPRQ
0000022BC4230010  56 48 31 D2 65 48 8B 52 60 48 8B 52 18 48 8B 52  VH1ÒeH.R`H.R.H.R
0000022BC4230020  20 48 8B 72 50 48 0F B7 4A 4A 4D 31 C9 48 31 C0   H.rPH.·JJM1ÉH1À
0000022BC4230030  AC 3C 61 7C 02 2C 20 41 C1 C9 0D 41 01 C1 E2 ED  ¬<a|., AÁÉ.A.Áâí
0000022BC4230040  52 41 51 48 8B 52 20 8B 42 3C 48 01 D0 8B 80 88  RAQH.R .B<H.Ð...
0000022BC4230050  00 00 00 48 85 C0 74 67 48 01 D0 50 8B 48 18 44  ...H.ÀtgH.ÐP.H.D
0000022BC4230060  8B 40 20 49 01 D0 E3 56 48 FF C9 41 8B 34 88 48  .@ I.ÐãVHÿÉA.4.H
0000022BC4230070  01 D6 4D 31 C9 48 31 C0 AC 41 C1 C9 0D 41 01 C1  .ÖM1ÉH1À¬AÁÉ.A.Á
0000022BC4230080  38 E0 75 F1 4C 03 4C 24 08 45 39 D1 75 D8 58 44  8àuñL.L$.E9ÑuØXD
0000022BC4230090  8B 40 24 49 01 D0 66 41 8B 0C 48 44 8B 40 1C 49  .@$I.ÐfA..HD.@.I
0000022BC42300A0  01 D0 41 8B 04 88 48 01 D0 41 58 41 58 5E 59 5A  .ÐA...H.ÐAXAX^YZ
0000022BC42300B0  41 58 41 59 41 5A 48 83 EC 20 41 52 FF E0 58 41  AXAYAZH.ì ARÿàXA
0000022BC42300C0  59 5A 48 8B 12 E9 57 FF FF FF 5D 48 BA 01 00 00  YZH..éWÿÿÿ]Hº...
0000022BC42300D0  00 00 00 00 00 48 8D 8D 01 01 00 00 41 BA 31 8B  .....H......Aº1.
0000022BC42300E0  6F 87 FF D5 BB E0 1D 2A 0A 41 BA A6 95 BD 9D FF  o.ÿÕ»à.*.Aº¦.½.ÿ
0000022BC42300F0  D5 48 83 C4 28 3C 06 7C 0A 80 FB E0 75 05 BB 47  ÕH.Ä(<.|..ûàu.»G
0000022BC4230100  13 72 6F 6A 00 59 41 89 DA FF D5 6E 6F 74 65 70  .roj.YA.ÚÿÕnotep
0000022BC4230110  61 64 2E 65 78 65 00 00 00 00 00 00 00 00 00 00  ad.exe..........
0000022BC4230120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000022BC4230130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

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