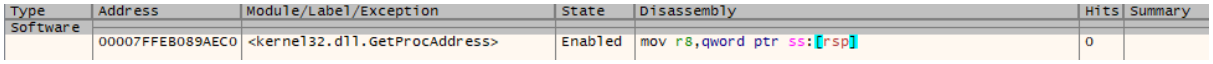
Reverse Engineering Obfuscated Function

In this section we will try to reverse engineer the Obfuscated Function, in the last part we had created an Obfuscated Function malware, so let's see whether we can reverse engineer it and see what functions it is calling.

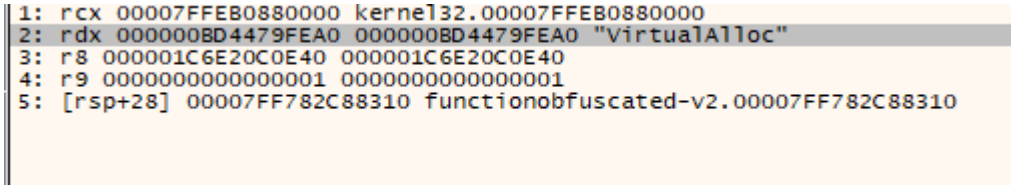
So, load the .exe file in xdbg, and now put the breakpoint in "GetProcAddress", because we know that it is calling that function, add the breakpoint, and we know that it takes two parameters, and in the 2nd parameter we have our function VirtualAlloc, so when we hit the breakpoint, so keep running the program, until you see VirtualAlloc.

Adding the breakpoint at GetProcAddress:



Type	Address	Module/Label/Exception	State	Disassembly	Hits	Summary
Software						
	00007FFEB089AEC0	<kernel32.dll.GetProcAddress>	Enabled	mov r8,qword ptr ss:[rsp]	0	

As we can he hit the breakpoint, and on analysing the 2nd parameter we get VirtualAlloc function:



```
1: rcx 00007FFEB0880000 kernel32.00007FFEB0880000
2: rdx 000000BD4479FEA0 000000BD4479FEA0 "VirtualAlloc"
3: r8 000001C6E20C0E40 000001C6E20C0E40
4: r9 0000000000000001 0000000000000001
5: [rsp+28] 00007FF782C88310 functionobfuscated-v2.00007FF782C88310
```