

1

0012FE5C	7C90DE5C	RETURN to ntdll.7C90DE5C
0012FE60	7C81CAB6	RETURN to kernel32.7C81CAB6
0012FE64	FFFFFFFF	
0012FE68	00000001	
0012FE6C	00000001	
0012FE70	FFFFFFFF	
0012FE74	7FFDB000	
0012FE78	00000000	
0012FE7C	00300014	
0012FE80	00000002	
0012FE84	00000580	
0012FE88	000004FC	
0012FE8C	0000C254	
0012FE90	00000000	
0012FE94	00000000	
0012FE98	00010003	
0012FE9C	00000000	
0012FEA0	7B7A7978	
0012FEA4	00000001	
0012FEA8	7C912CAE	RETURN to ntdll.7C912CAE from ntdll.RtlEnterCriticalSection
0012FEAC	7C912CE4	RETURN to ntdll.7C912CE4 from ntdll.7C90E8E6
0012FEB0	7C912D51	RETURN to ntdll.7C912D51 from ntdll.RtlLeaveCriticalSection
0012FEB4	7C912D58	RETURN to ntdll.7C912D58 from ntdll.7C90E8E6
0012FEB8	00000006	
0012FEBE	00241EE0	
0012FEC0	00000208	
0012FEC4	9F9E9D9C	
0012FEC8	0012FE88	
0012FECC	A7A6A5A4	
0012FED0	0012FF28	
0012FED4	00000000	
0012FED8	00000000	
0012FEDC	FFFFFFFF	

Il valore del parametro *CommandLine* è memorizzato nella memoria dello stack all'indirizzo *0x0012FE5C*. Questo valore è una stringa di 14 caratteri che rappresenta un messaggio che viene visualizzato quando il codice viene eseguito in modo anomalo.

2

Registers (FPU)	
ERX	0A280105
ECX	7FFD6000
EDX	00000A28
EBX	7FFD6000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.7C910208
EIP	004015A3 Malware_.004015A3
C 0	ES 0023 32bit 0(FFFFFFFF)
P 1	CS 001B 32bit 0(FFFFFFFF)
A 0	SS 0023 32bit 0(FFFFFFFF)
Z 0	DS 0023 32bit 0(FFFFFFFF)
S 0	FS 003B 32bit 7FFDF000(FFF)
T 0	GS 0000 NULL
D 0	0 0 LastErr ERROR_INVALID_HANDLE (00000006)
EFL	00000206 (NO,NB,NE,A,NS,PE,GE,G)
ST0	empty -UNORM BCBC 01050104 0054005C
ST1	empty +UNORM 0061 0072006F 00070060
ST2	empty +UNORM 0063 00650072 00690044
ST3	empty +UNORM 0066 00200034 00200079
ST4	empty +UNORM 0073 006E0061 00720074
ST5	empty +UNORM 006F 006C0063 0070002E
ST6	empty +UNORM 0062 0073006A 004C0070
ST7	empty +UNORM 005C 00700069 007A002E
	3 2 1 0 E S P U O Z D I
FST 0000	Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCW 027F	Prec NEAR,53 Mask 1 1 1 1 1 1

Il valore del registro EDX è *00000A28*.

3 - 4 - 5

The screenshot shows a debugger interface with two main panes. The left pane displays assembly code with several instructions highlighted in yellow. One instruction, at address 00401595, is specifically highlighted in red. The right pane shows the 'Registers (FPU)' window with various CPU registers listed and their current values.

Register	Value
EDX	00228105
ECX	7FFD6000
EDI	00000000
EDP	00000000
ESP	0012FF94
EBP	0012FFC0
ESI	000000FF
EDI	7C910208 nt.dll.7C910208
EIP	004015A5 Malware_..004015A5

The assembly code in the left pane includes:

```

00401559: F7D9    NEG ECX
00401560: 4F      DEC EDI
00401561: B845 00  MOU AL,BYTE PTR SS:[EBP+C1]
00401562: 48      STO EDI
00401563: F2:E8  REPNE SCAS BYTE PTR ES:[EDI]
00401564: 47      INC EDI
00401565: C707 00  CMPSB PTR DS:[EDI],AL
00401566: 74 04  JE SHORT Malware_..00401571
00401567: 33C0  XOR EAX,EAX
00401568: 33D8 02  XOR EDX,EDX
00401569: > FC    JMP SHORT Malware_..00401573
00401570: 8B07    MOU EDX,EDI
00401571: C3    RETN
00401572: F5      LEAVE
00401573: C3    RETN
00401574: S5      PUSH EBP
00401575: 48 EC  MOU EDI,ESP
00401576: 6A FF  PUSH -1
00401577: 68 C0404000 PUSH Malware_..004040C0
00401578: 3C25400000 MOU EDI,004040C0
00401579: 641H 00000000 MOU EDX,00000000 PTR FS:[0]
00401580: S8      PUSH EBX
00401581: 6418925 000000 MOU DWORD PTR FS:[0],ESP
00401582: C1E8 10  SHL ECX,10
00401583: S8      PUSH EBX
00401584: S865 E8  MOU DWORD PTR SS:[EBP-18],ESP
00401585: FF15 30404000 CALL DWORD PTR DS:[<&KERNEL32.GetVersion] kernel32.GetVersion
00401586: 8004    MOU EDX,00000000
00401587: 8915 D4524000 MOU DWORD PTR DS:[4052D4],EDX
00401588: 88C8    MOU ECX,EDX
00401589: FF00000000 MOU EDI,00000000
0040158B: 8900 D0524000 MOU DWORD PTR DS:[4052D01],ECX
0040158C: C1E1 08  SHL ECX,8
0040158D: 8900 CC524000 MOU DWORD PTR DS:[4052CC1],ECX
0040158E: C1E8 10  SHR EDX,10
0040158F: 8900 C0524000 MOU DWORD PTR DS:[4052C81],EXX
00401590: 6A 1C  PUSH IC
00401591: E8 33090000 CALL Malware_..00401F08
00401592: F9      POP ECX
00401593: 8900 C0524000 MOU DWORD PTR DS:[4052C81],EXX
00401594: > 75 08  JNZ SHORT Malware_..004015E2
00401595: 6A 1C  PUSH IC
00401596: E8 30090000 CALL Malware_..0040167B
00401597: F9      POP ECX
00401598: 8965 FC 00  AND DWORD PTR SS:[EBP+4],0
00401599: C707 00000000 CMPSB PTR DS:[EBP+C1],0
0040159A: FF15 30404000 CALL DWORD PTR DS:[<&KERNEL32.GetCommandLine] GetCommandLineA
0040159B: A9 D0574000 MOU DWORD PTR DS:[4057D01],EXX
0040159C: E8 30090000 CALL Malware_..0040167B
0040159D: 8905240000 MOU DWORD PTR DS:[4052401],EXX
0040159E: E8 D9030000 CALL Malware_..0040193D
0040159F: E8 1B030000 CALL Malware_..00401925
00401600: E8 90000000 CALL Malware_..0040169F

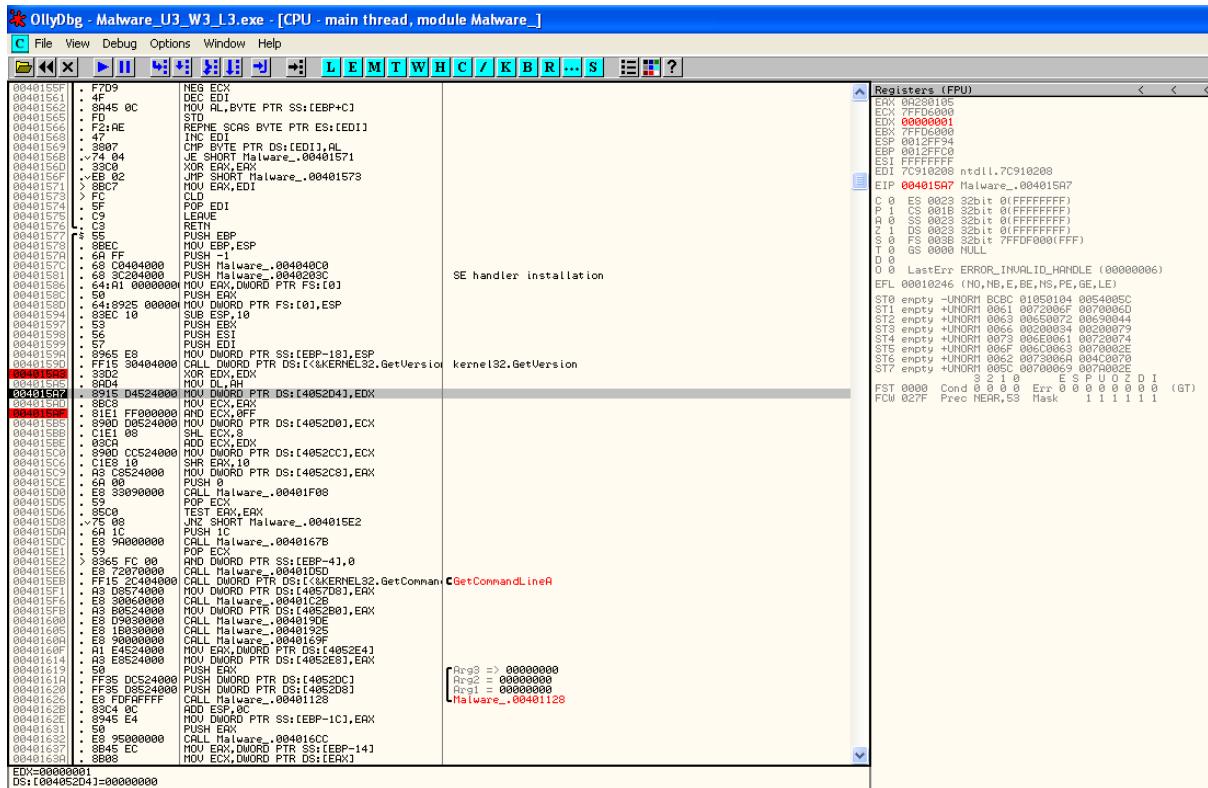
```

Dopo aver eseguito uno *step-into* dal *breakpoint* impostato, il valore del registro *EDX* è ora 0. Questo perché il breakpoint è stato impostato sulla linea di codice che inizializza il registro *EDX* a 0. L'istruzione che è stata eseguita è

mov edx, 0

Essa sposta il valore 0 nel registro *EDX*. Quando si esegue uno *step-into*, il *debugger* passa alla successiva istruzione di codice. In questo caso, l'istruzione successiva è quella che ho appena messo in evidenza. Pertanto, il valore del registro *EDX* è ora 0.

Dopo aver impostato un secondo *breakpoint* all'indirizzo di memoria **004015AF**, il valore del registro **ECX** è **7FFD6000**.



Dopo aver eseguito uno *step-into* a partire dal secondo *breakpoint*, il valore del registro *ECX* è 00000001. L'istruzione eseguita è

MOV DWORD PTR DS: [4052D4], EDX

Bonus

Potrebbe essere un ransomware, ma non ne sono sicuro.