

OllyDBG

L'esercizio di oggi ci chiedeva, prendendo come riferimento il Malware_U3_W3_L3, di rispondere ai seguenti quesiti:

1. All'indirizzo 0040106E il Malware effettua una chiamata di funzione alla funzione "CreateProcess". Qual è il valore del parametro "CommandLine" passato sullo stack?
2. Inserire un breakpoint software all'indirizzo 004015A3, Qual è il valore del registro EDX, eseguire poi uno "step-into" e indicare il nuovo valore di EDX motivando la risposta e indicando l'istruzione eseguita.
3. Inserire un secondo breakpoint all'indirizzo 004015AF, indicare il valore del registro ECX, eseguire poi uno step-into e indicare il nuovo valore ECX spiegando l'istruzione eseguita.

1.Valore “CommandLine”

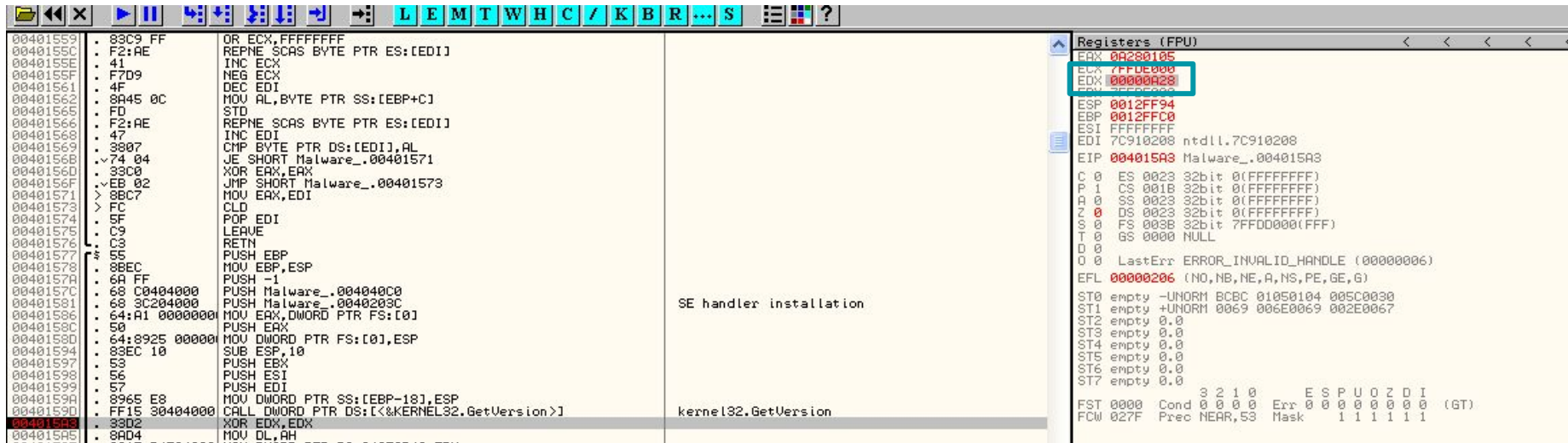
Il primo punto dell’esercizio ci chiedeva di trovare il valore del parametro “CommandLine” che viene passato sullo stack.

Nella figura qua sotto possiamo vedere che il valore del parametro riportato sullo stack di “**CommandLine**” è “**cmd**”.

0040105B	. 6A 00	PUSH 0	CurrentDir = NULL
0040105D	. 6A 00	PUSH 0	pEnvironment = NULL
0040105F	. 6A 00	PUSH 0	CreationFlags = 0
00401061	. 6A 01	PUSH 1	InheritHandles = TRUE
00401063	. 6A 00	PUSH 0	pThreadSecurity = NULL
00401065	. 6A 00	PUSH 0	pProcessSecurity = NULL
00401067	. 68 30504000	PUSH Malware_.00405030	CommandLine = "cmd"
0040106C	. 6A 00	PUSH 0	ModuleFileName = NULL
0040106E	. FF15 04404000	CALL DWORD PTR DS:[&KERNEL32.CreateProcessA]	CreateProcessA
00401074	. 004E EC	CALL DWORD PTR DS:[&FEDD_141 Env]	

2. Inserire Breakpoint-EDX-

Il secondo punto ci chiedeva di inserire il breakpoint all'indirizzo 004015A3, inserire il valore iniziale di EDX e il suo valore dopo aver eseguito lo step-into e che istruzione è stata eseguita. Dopo aver creato il nostro breakpoint, possiamo vedere che il valore di EDX è 00000A28.



The screenshot displays a debugger interface with two main panels. The left panel shows assembly code with addresses from 00401559 to 004015A5. The right panel shows the 'Registers (FPU)' window.

Assembly Code (Left Panel):

```
00401559 . 83C9 FF      OR ECX,FFFFFFFF
0040155C . F2:AE        REPNE SCAS BYTE PTR ES:[EDI]
0040155E . 41           INC ECX
0040155F . F7D9         NEG ECX
00401561 . 4F           DEC EDI
00401562 . 8A45 0C      MOV AL,BYTE PTR SS:[EBP+C]
00401565 . FD          STD
00401566 . F2:AE        REPNE SCAS BYTE PTR ES:[EDI]
00401568 . 47           INC EDI
00401569 . 3807         CMP BYTE PTR DS:[EDI],AL
0040156B . 74 04        JE SHORT Malware_.00401571
0040156D . 33C0         XOR EAX,EAX
0040156F . EB 02        JMP SHORT Malware_.00401573
00401571 . 8BC7        MOV EAX,EDI
00401573 . FC          CLD
00401574 . 5F          POP EDI
00401575 . C9          LEAVE
00401576 . C3          RETN
00401577 . 55          PUSH EBP
00401578 . 8BEC        MOV EBP,ESP
0040157A . 6A FF        PUSH -1
0040157C . 68 C0404000 PUSH Malware_.004040C0
00401581 . 68 3C204000 PUSH Malware_.0040203C
00401586 . 64:A1 00000000 MOV EAX,DWORD PTR FS:[0]
0040158C . 50          PUSH EAX
0040158D . 64:8925 00000000 MOV DWORD PTR FS:[0],ESP
00401594 . 83EC 10      SUB ESP,10
00401597 . 53          PUSH EBX
00401598 . 56          PUSH ESI
00401599 . 57          PUSH EDI
0040159A . 965 E8      MOV DWORD PTR SS:[EBP-18],ESP
0040159D . FF15 30404000 CALL DWORD PTR DS:[<&KERNEL32.GetVersion>]
004015A3 . 33D2        XOR EDX,EDX
004015A5 . 8A04        MOV DL,AH
```

Registers (FPU) Window (Right Panel):

Register	Value
EAX	00280105
ECX	7FDE0000
EDX	00000A28
ESI	FFFFFFF0
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 7FDD0000(FFF)
T 0	GS 0000 NULL
D 0	
Q 0	LastErr ERROR_INVALID_HANDLE (00000006)
EFL	00000206 (NO,NB,NE,A,NS,PE,GE,G)
ST0	empty -UNORM BCBC 01050104 005C0030
ST1	empty +UNORM 0069 006E0069 002E0067
ST2	empty 0.0
ST3	empty 0.0
ST4	empty 0.0
ST5	empty 0.0
ST6	empty 0.0
ST7	empty 0.0
FST	0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 (GT)
FCW	027F Prec NEAR,53 Mask 1 1 1 1 1 1

Dopo invece aver eseguito “step-into” possiamo vedere come il valore EDX cambia, diventando 00000000. Questo perchè XOR azzerava il registro, (EDX-EDX)restituendoci quindi questo nuovo valore.

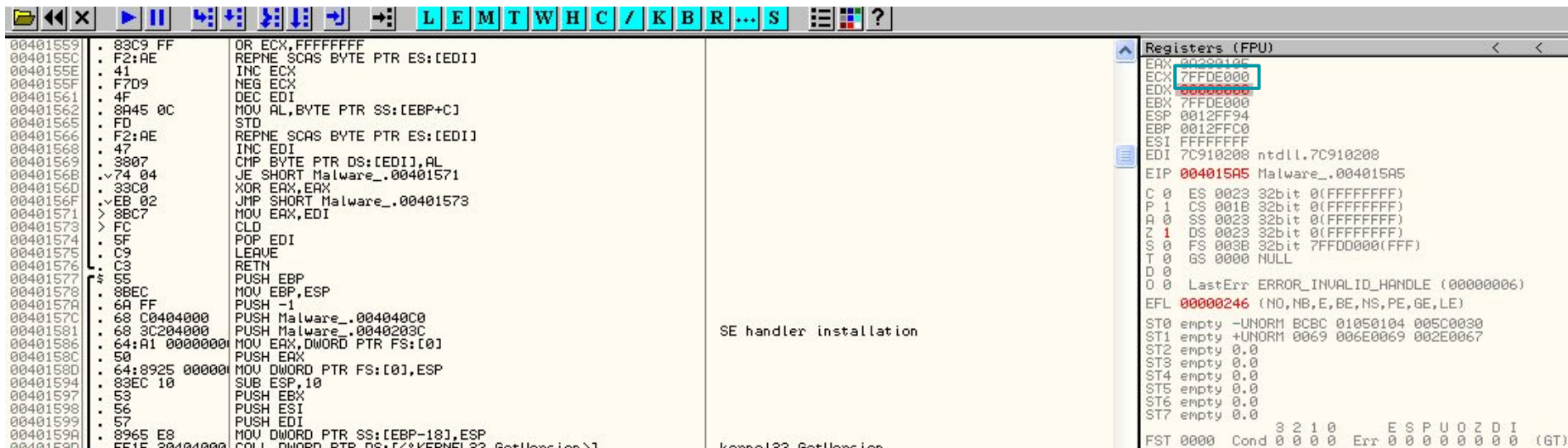
The screenshot shows a debugger interface with two main panes. The left pane displays assembly code with the following instructions highlighted in red:

- 00401559: 83C9 FF OR ECX,FFFFFFFF
- 0040155C: F2:AE REPNE SCAS BYTE PTR ES:[EDI]
- 0040155E: 41 INC ECX
- 0040155F: F7D9 NEG ECX
- 00401561: 4F DEC EDI
- 00401562: 8A45 0C MOV AL, BYTE PTR SS:[EBP+C]
- 00401565: FD STD
- 00401566: F2:AE REPNE SCAS BYTE PTR ES:[EDI]
- 00401568: 47 INC EDI
- 00401569: 3B07 CMP BYTE PTR DS:[EDI],AL
- 0040156B: 74 04 JE SHORT Malware_.00401571
- 0040156D: 33C0 XOR EAX,EAX
- 0040156F: EB 02 JMP SHORT Malware_.00401573
- 00401571: 8BC7 MOV EAX,EDI
- 00401573: FC CLD
- 00401574: 5F POP EDI
- 00401575: C9 LEAVE
- 00401576: C3 RETN
- 00401577: 55 PUSH EBP
- 00401578: 8BEC MOV EBP,ESP
- 0040157A: 6A FF PUSH -1
- 0040157C: 68 C0404000 PUSH Malware_.004040C0
- 00401581: 68 3C204000 PUSH Malware_.0040203C
- 00401586: 64:A1 00000000 MOV EAX,DWORD PTR FS:[0]
- 0040158C: 50 PUSH EAX
- 0040158D: 64:8925 00000000 MOV DWORD PTR FS:[0],ESP
- 00401594: 83EC 10 SUB ESP,10
- 00401597: 53 PUSH EBX
- 00401598: 56 PUSH ESI
- 00401599: 57 PUSH EDI
- 0040159A: 8965 E8 MOV DWORD PTR SS:[EBP-18],ESP
- 0040159D: FF15 30404000 CALL DWORD PTR DS:[<kernel32.GetVersion>]
- 004015A0: 33D2 XOR EDX,EDX

The right pane shows the "Registers (FPU)" window. The EDX register is highlighted with a red box and shows the value 00000000. Other registers like EAX, ECX, ESP, EBP, ESI, EDI, EIP, C, P, A, Z, S, T, D, O, EFL, ST0, ST1, ST2, ST3, ST4, ST5, ST6, ST7, FST, and FCW are also visible.

3. Inserire Breakpoint -ECX-

Il terzo punto ci chiedeva di inserire il breakpoint all'indirizzo 004015AF, inserire il valore iniziale di ECX e il suo valore dopo aver eseguito lo step-into e che istruzione è stata eseguita. Dopo aver creato il nostro breakpoint, possiamo vedere che il valore di ECX è 7FFDE000.



The screenshot shows a debugger window with the following components:

- Assembly View:** Displays assembly instructions with their addresses. The instruction at address 004015AF is `OR ECX,FFFFFFFF`. The instruction at address 004015B5 is `SE handler installation`.
- Registers (FPU) View:** Shows the current values of the registers. The `ECX` register is highlighted with a red box and contains the value `7FFDE000`.

```
00401559 . 83C9 FF OR ECX,FFFFFFFF
0040155C . F2:AE REPNE SCAS BYTE PTR ES:[EDI]
0040155E . 41 INC ECX
0040155F . F7D9 NEG ECX
00401561 . 4F DEC EDI
00401562 . 8A45 0C MOV AL,BYTE PTR SS:[EBP+C]
00401565 . FD STD
00401566 . F2:AE REPNE SCAS BYTE PTR ES:[EDI]
00401568 . 47 INC EDI
00401569 . 3B07 CMP BYTE PTR DS:[EDI],AL
0040156B . 74 04 JE SHORT Malware_.00401571
0040156D . 33C0 XOR EAX,EAX
0040156F . EB 02 JMP SHORT Malware_.00401573
00401571 . 8BC7 MOV EAX,EDI
00401573 . FC CLD
00401574 . 5F POP EDI
00401575 . C3 LEAVE
00401576 . C3 RETN
00401577 . 55 PUSH EBP
00401578 . 8BEC MOV EBP,ESP
0040157A . 6A FF PUSH -1
0040157C . 68 C0404000 PUSH Malware_.004040C0
00401581 . 68 3C204000 PUSH Malware_.0040203C
00401586 . 64:A1 00000000 MOV EAX,DWORD PTR FS:[0]
0040158C . 50 PUSH EAX
0040158D . 64:8925 00000000 MOV DWORD PTR FS:[0],ESP
00401594 . 33EC 10 SUB ESP,10
00401597 . 53 PUSH EBX
00401598 . 56 PUSH ESI
00401599 . 57 PUSH EDI
0040159A . 8965 E8 MOV DWORD PTR SS:[EBP-18],ESP
0040159D . 55 LEAVE
```

Registers (FPU)

Register	Value
EAX	00000000
ECX	7FFDE000
EDX	00000000
EBX	7FFDE000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.7C910208
EIP	004015A5 Malware_.004015A5
C 0	ES 0023 32bit 0(FFFFFFFF)
P 1	CS 001B 32bit 0(FFFFFFFF)
A 0	SS 0023 32bit 0(FFFFFFFF)
Z 1	DS 0023 32bit 0(FFFFFFFF)
S 0	FS 003B 32bit 7FFDD000(FFF)
T 0	GS 0000 NULL
D 0	
O 0	LastErr ERROR_INVALID_HANDLE (00000006)
EFL	00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0	empty -UNORM BCBC 01050104 005C0030
ST1	empty +UNORM 0069 006E0069 002E0067
ST2	empty 0.0
ST3	empty 0.0
ST4	empty 0.0
ST5	empty 0.0
ST6	empty 0.0
ST7	empty 0.0
FST	0000 Cond 3 2 1 0 Err 0 0 0 0 0 0 0 0 (GT)

Dopo invece aver eseguito “step-into” possiamo vedere come il valore ECX cambia, diventando 00000005. Questo perchè AND fa una moltiplicazione tra ECX e 0FF, restituendoci quindi questo nuovo valore.

Address	Disassembly	Comment
00401577	55	PUSH EBP
00401578	8BEC	MOV EBP,ESP
00401579	6A FF	PUSH -1
0040157C	68 C0404000	PUSH Malware_.004040C0
00401581	68 3C204000	PUSH Malware_.0040203C
00401586	64:R1 00000000	MOV EAX,DWORD PTR FS:[0]
0040158C	50	PUSH EAX
0040158D	64:8925 00000000	MOV DWORD PTR FS:[0],ESP
00401594	83EC 10	SUB ESP,10
00401597	53	PUSH EBX
00401598	54	PUSH ESI
00401599	56	PUSH EDI
0040159A	8965 E8	MOV DWORD PTR SS:[EBP-18],ESP
0040159D	FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion>]
004015A0	33D2	XOR EDX,EDX
004015A5	8A04	MOV DL,AH
004015A7	8915 04524000	MOV DWORD PTR DS:[405204],EDX
004015AD	8BC8	MOV ECX,ECX
004015B0	81E1 FF000000	AND ECX,0FF
004015B5	8900 00524000	MOV DWORD PTR DS:[405200],ECX
004015B8	C1E1 08	SHL ECX,8
004015BE	03CA	ADD ECX,EDX
004015C0	890D CC524000	MOV DWORD PTR DS:[4052CC],ECX
004015C6	C1E8 10	SHR EAX,10
004015C9	A3 C8524000	MOV DWORD PTR DS:[4052C8],EAX
004015CE	6A 00	PUSH 0
004015D0	E8 33090000	CALL Malware_.00401F08
004015D5	59	POP ECX
004015D6	85C0	TEST EAX,EAX
004015D8	75 08	JNZ SHORT Malware_.004015E2
004015DA	6A 1C	PUSH 1C
004015DC	E8 9A000000	CALL Malware_.0040167B
004015E1	59	POP ECX
004015E3	890D CC524000	MOV DWORD PTR DS:[4052CC],ECX

Register	Value
EAX	00000000
ECX	00000005
EDX	00000000
EBX	7FFDC000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.7C910208
EIP	004015B5 Malware_.004015B5
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	
O 0	LastErr ERROR_INVALID_HANDLE (00000006)
EFL	00000206 (NO,NB,NE,A,NS,PE,GE,G)
ST0	empty -UNORM BCBC 01050104 005C0030
ST1	empty +UNORM 0069 006E0069 002E0067
ST2	empty 0.0
ST3	empty 0.0
ST4	empty 0.0
ST5	empty 0.0
ST6	empty 0.0
ST7	empty 0.0
FST	0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCW	027F Prec NEAR,S3 Mask 1 1 1 1 1 1