

EPICODE

CYBERSECURITY COURSE

BY MAX ALDROVANDI

17/05/2024

Web-site: <https://epicode.com/it>

Locality:

Via dei Magazzini Generali,

6 Roma, Lazio 00154, IT

PRACTICE EXERCISE S11/L3

Track:

Refer to the malware: **Malware_U3_W3_L3**, found inside the **Exercise_Practice_U3_W3_L3** folder on the desktop of the virtual machine dedicated to malware analysis. Answer the following questions using OllyDBG.

- At address **0040106E** the **Malwareeffects** a function call to the "**CreateProcess**" function. What is the value of the "**CommandLine**" parameter that is passed on the stack?
- Enter a **breakpointsoftware** at address **004015A3**. What is the value of the EDX register? Perform a "step-into" at this point. Indicate what the value of the EDX register is now by giving reasons for your answer. What instruction was executed?
- Enter a second breakpoint at memory address **004015AF**. What is the value of the ECX register? Execute a step-into. What is now the value of ECX? Explain which instruction was executed.
- **BONUS:** Explain in broad terms how the malware works.

Solution:

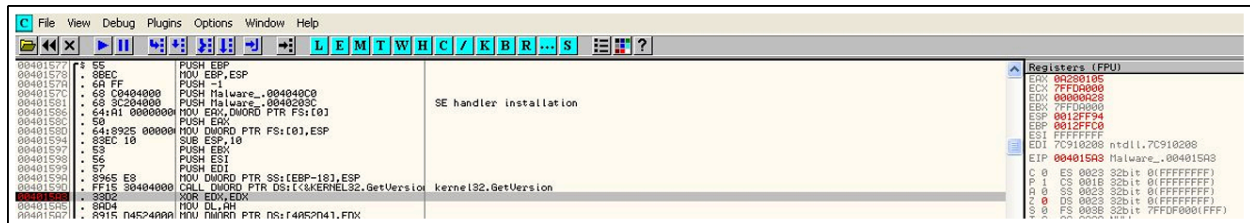
The value of the parameter is "CMD" or the Windows command prompt, as seen in the figure below at 00401067.

00401057	. 8045 A8	LEA EAX,DWORD PTR SS:[EBP-58]	
0040105A	. 50	PUSH EAX	
0040105B	. 6A 00	PUSH 0	
0040105D	. 6A 00	PUSH 0	
0040105F	. 6A 00	PUSH 0	
00401061	. 6A 01	PUSH 1	
00401063	. 6A 00	PUSH 0	
00401065	. 6A 00	PUSH 0	
00401067	. 68 30504000	PUSH Malware_.00405030	
0040106C	. 6A 00	PUSH 0	
0040106E	. FF15 04404000	CALL DWORD PTR DS:[<&KERNEL32.CreateProcessA]	pStartupInfo CurrentDir = NULL pEnvironment = NULL CreationFlags = 0 InheritHandles = TRUE pThreadSecurity = NULL pProcessSecurity = NULL CommandLine = "cmd" ModuleFileName = NULL CreateProcessA
00401074	. 8945 EC	MOV DWORD PTR SS:[EBP-14],EAX	
00401077	. 6A FF	PUSH -1	
00401079	. 8B4D F0	MOV ECX,DWORD PTR SS:[EBP-10]	
0040107C	. 51	PUSH ECX	
0040107D	. FF15 00404000	CALL DWORD PTR DS:[<&KERNEL32.WaitForSingleObject]	Timeout = INFINITE hObject WaitForSingleObject
00401083	. 33C0	XOR EAX,EAX	
00401085	. 8BE5	MOV ESP,EBP	
00401087	. 5D	POP EBP	
00401089	. C2	RETN	

Once the breakpoint is configured, we click on "play", the program will stop at instruction **XOR EDX,EDX**. Before the instruction is executed, the register value is "**00000A28**".

After the step-into, the instruction **XOR EDX,EDX** is executed, which in fact is equivalent to initializing a variable to zero.

Thus, after step-into, the value of **EDX** will be 0.

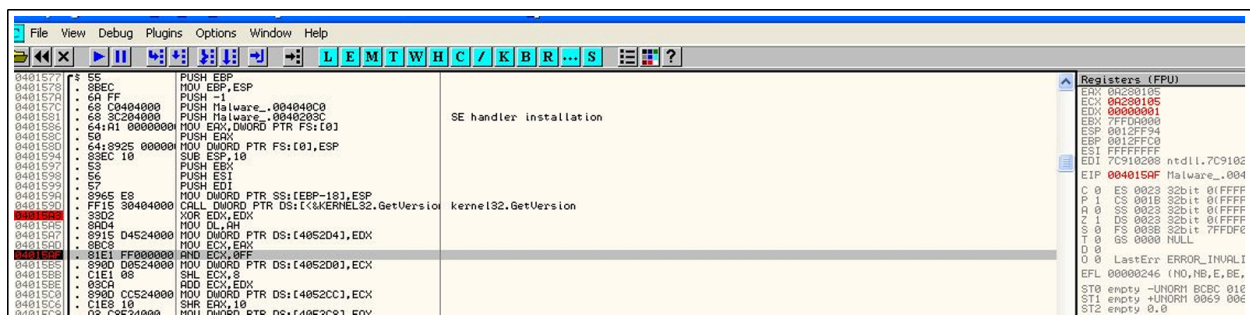


Before



After

We configure the second breakpoint. The value of the **ECX** register is "**0A280105**".



Before

OllyDbg - Malware_U3_W3_1.3.exe - [CPU - main thread, module Malware_]

File View Debug Plugins Options Window Help

SE handler installation

```

00401577 65      PUSH ESP
00401578 66      MOV EIP, ESP
00401579 6A FF    PUSH -1
0040157C 68 00404000  PUSH Malware_00404000
00401580 68 00404000  PUSH Malware_00404000
00401581 64 00000000  MOV EDI, DWORD PTR FS:[0]
00401582 68      PUSH ESI
00401583 64 00000000  MOV EDI, DWORD PTR FS:[0], ESI
00401584 66      SUB ESP, 10
00401585 65      PUSH EBP
00401586 66      MOV EBP, ESP
00401587 65      PUSH ESI
00401588 66      MOV ESI, EBP
00401589 67      PUSH EDI
0040158A 68      MOV EDI, DWORD PTR SS:[EBP-19], ESI
0040158B 68 00404000  CALL DWORD PTR DS:[<kernel32.GetVersion> kernel32.GetVersion
0040158C 68      MOV EDI, EAX
0040158D 68      MOV EAX, EDI
0040158E 68      MOV EDI, EAX
0040158F 68      MOV EAX, EDI
00401590 68      MOV EAX, EDI
00401591 68      MOV EAX, EDI
00401592 68      MOV EAX, EDI
00401593 68      MOV EAX, EDI
00401594 68      MOV EAX, EDI
00401595 68      MOV EAX, EDI
00401596 68      MOV EAX, EDI
00401597 68      MOV EAX, EDI
00401598 68      MOV EAX, EDI
00401599 68      MOV EAX, EDI
0040159A 68      MOV EAX, EDI
0040159B 68      MOV EAX, EDI
0040159C 68      MOV EAX, EDI
0040159D 68      MOV EAX, EDI
0040159E 68      MOV EAX, EDI
0040159F 68      MOV EAX, EDI
004015A0 68      MOV EAX, EDI
004015A1 68      MOV EAX, EDI
004015A2 68      MOV EAX, EDI
004015A3 68      MOV EAX, EDI
004015A4 68      MOV EAX, EDI
004015A5 68      MOV EAX, EDI
004015A6 68      MOV EAX, EDI
004015A7 68      MOV EAX, EDI
004015A8 68      MOV EAX, EDI
004015A9 68      MOV EAX, EDI
004015AA 68      MOV EAX, EDI
004015AB 68      MOV EAX, EDI
004015AC 68      MOV EAX, EDI
004015AD 68      MOV EAX, EDI
004015AE 68      MOV EAX, EDI
004015AF 68      MOV EAX, EDI
004015B0 68      MOV EAX, EDI
004015B1 68      MOV EAX, EDI
004015B2 68      MOV EAX, EDI
004015B3 68      MOV EAX, EDI
004015B4 68      MOV EAX, EDI
004015B5 68      MOV EAX, EDI
004015B6 68      MOV EAX, EDI
004015B7 68      MOV EAX, EDI
004015B8 68      MOV EAX, EDI
004015B9 68      MOV EAX, EDI
004015BA 68      MOV EAX, EDI
004015BB 68      MOV EAX, EDI
004015BC 68      MOV EAX, EDI
004015BD 68      MOV EAX, EDI
004015BE 68      MOV EAX, EDI
004015BF 68      MOV EAX, EDI
004015C0 68      MOV EAX, EDI
004015C1 68      MOV EAX, EDI
004015C2 68      MOV EAX, EDI
004015C3 68      MOV EAX, EDI
004015C4 68      MOV EAX, EDI
004015C5 68      MOV EAX, EDI
004015C6 68      MOV EAX, EDI
004015C7 68      MOV EAX, EDI
004015C8 68      MOV EAX, EDI
004015C9 68      MOV EAX, EDI
004015CA 68      MOV EAX, EDI
004015CB 68      MOV EAX, EDI
004015CC 68      MOV EAX, EDI
004015CD 68      MOV EAX, EDI
004015CE 68      MOV EAX, EDI
004015CF 68      MOV EAX, EDI
004015D0 68      MOV EAX, EDI
004015D1 68      MOV EAX, EDI
004015D2 68      MOV EAX, EDI
004015D3 68      MOV EAX, EDI
004015D4 68      MOV EAX, EDI
004015D5 68      MOV EAX, EDI
004015D6 68      MOV EAX, EDI
004015D7 68      MOV EAX, EDI
004015D8 68      MOV EAX, EDI
004015D9 68      MOV EAX, EDI
004015DA 68      MOV EAX, EDI
004015DB 68      MOV EAX, EDI
004015DC 68      MOV EAX, EDI
004015DD 68      MOV EAX, EDI
004015DE 68      MOV EAX, EDI
004015DF 68      MOV EAX, EDI
004015E0 68      MOV EAX, EDI
004015E1 68      MOV EAX, EDI
004015E2 68      MOV EAX, EDI
004015E3 68      MOV EAX, EDI
004015E4 68      MOV EAX, EDI
004015E5 68      MOV EAX, EDI
004015E6 68      MOV EAX, EDI
004015E7 68      MOV EAX, EDI
004015E8 68      MOV EAX, EDI
004015E9 68      MOV EAX, EDI
004015EA 68      MOV EAX, EDI
004015EB 68      MOV EAX, EDI
004015EC 68      MOV EAX, EDI
004015ED 68      MOV EAX, EDI
004015EE 68      MOV EAX, EDI
004015EF 68      MOV EAX, EDI
004015F0 68      MOV EAX, EDI
004015F1 68      MOV EAX, EDI
004015F2 68      MOV EAX, EDI
004015F3 68      MOV EAX, EDI
004015F4 68      MOV EAX, EDI
004015F5 68      MOV EAX, EDI
004015F6 68      MOV EAX, EDI
004015F7 68      MOV EAX, EDI
004015F8 68      MOV EAX, EDI
004015F9 68      MOV EAX, EDI
004015FA 68      MOV EAX, EDI
004015FB 68      MOV EAX, EDI
004015FC 68      MOV EAX, EDI
004015FD 68      MOV EAX, EDI
004015FE 68      MOV EAX, EDI
004015FF 68      MOV EAX, EDI
00401600 68      MOV EAX, EDI
00401601 68      MOV EAX, EDI
00401602 68      MOV EAX, EDI
00401603 68      MOV EAX, EDI
00401604 68      MOV EAX, EDI
00401605 68      MOV EAX, EDI
00401606 68      MOV EAX, EDI
00401607 68      MOV EAX, EDI
00401608 68      MOV EAX, EDI
00401609 68      MOV EAX, EDI
0040160A 68      MOV EAX, EDI
0040160B 68      MOV EAX, EDI
0040160C 68      MOV EAX, EDI
0040160D 68      MOV EAX, EDI
0040160E 68      MOV EAX, EDI
0040160F 68      MOV EAX, EDI
00401610 68      MOV EAX, EDI
00401611 68      MOV EAX, EDI
00401612 68      MOV EAX, EDI
00401613 68      MOV EAX, EDI
00401614 68      MOV EAX, EDI
00401615 68      MOV EAX, EDI
00401616 68      MOV EAX, EDI
00401617 68      MOV EAX, EDI
00401618 68      MOV EAX, EDI
00401619 68      MOV EAX, EDI
0040161A 68      MOV EAX, EDI
0040161B 68      MOV EAX, EDI
0040161C 68      MOV EAX, EDI
0040161D 68      MOV EAX, EDI
0040161E 68      MOV EAX, EDI
0040161F 68      MOV EAX, EDI
00401620 68      MOV EAX, EDI
00401621 68      MOV EAX, EDI
00401622 68      MOV EAX, EDI
00401623 68      MOV EAX, EDI
00401624 68      MOV EAX, EDI
00401625 68      MOV EAX, EDI
00401626 68      MOV EAX, EDI
00401627 68      MOV EAX, EDI
00401628 68      MOV EAX, EDI
00401629 68      MOV EAX, EDI
0040162A 68      MOV EAX, EDI
0040162B 68      MOV EAX, EDI
0040162C 68      MOV EAX, EDI
0040162D 68      MOV EAX, EDI
0040162E 68      MOV EAX, EDI
0040162F 68      MOV EAX, EDI
00401630 68      MOV EAX, EDI
00401631 68      MOV EAX, EDI
00401632 68      MOV EAX, EDI
00401633 68      MOV EAX, EDI
00401634 68      MOV EAX, EDI
00401635 68      MOV EAX, EDI
00401636 68      MOV EAX, EDI
00401637 68      MOV EAX, EDI
00401638 68      MOV EAX, EDI
00401639 68      MOV EAX, EDI
0040163A 68      MOV EAX, EDI
0040163B 68      MOV EAX, EDI
0040163C 68      MOV EAX, EDI
0040163D 68      MOV EAX, EDI
0040163E 68      MOV EAX, EDI
0040163F 68      MOV EAX, EDI
00401640 68      MOV EAX, EDI
00401641 68      MOV EAX, EDI
00401642 68      MOV EAX, EDI
00401643 68      MOV EAX, EDI
00401644 68      MOV EAX, EDI
00401645 68      MOV EAX, EDI
00401646 68      MOV EAX, EDI
00401647 68      MOV EAX, EDI
00401648 68      MOV EAX, EDI
00401649 68
```

In detail, the instruction performs logical **AND** on the bits of **EAX** and the hexadecimal value **FF**. We first bring both values into binary format and then perform logical **AND** between the bits.

Which in **Hexadecimal** is 00000005

4