



Unità 3 settimana 3 compito 3

1° ESERCIZIO

Il valore del parametro è "CMD", che rappresenta il prompt dei comandi di Windows. Questo è evidente nella figura che mostra l'indirizzo 00401067.

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



2° ESERCIZIO

Una volta impostato il breakpoint, clicchiamo su "play"; il programma si interromperà all'istruzione XOR EDX, EDX. Prima dell'esecuzione di questa istruzione, il valore del registro è "00000A28". Dopo l'esecuzione dell'istruzione "step-into", la XOR tra EDX e se stesso comporta l'inizializzazione a zero della variabile. Pertanto, dopo lo step-into, il valore di EDX sarà 0.

PRIMA:

Address	Disassembly	Comment
00401577	PUSH EBP	
00401578	MOV EBP, ESP	
0040157A	PUSH -1	
0040157C	PUSH Halware_.004040C0	
00401581	PUSH Halware_.0040203C	SE handler installation
00401584	MOV EAX, DWORD PTR FS:[0]	
0040158C	PUSH EAX	
00401590	MOV DWORD PTR FS:[0], ESP	
00401594	SUB ESP, 10	
00401597	PUSH EBX	
00401598	PUSH ESI	
00401599	PUSH EDI	
0040159A	MOV DWORD PTR SS:[EBP-10], ESP	
0040159D	CALL DWORD PTR DS:[<KERNEL32.GetVersion	kernel32.GetVersion
004015A2	XOR EDX, EDX	
004015A3	MOV DL, AH	
004015A7	MOV DWORD PTR DS:[405204], EAX	

Register	Value
EAX	00200105
ECK	7FFD0000
EDI	00000000
EBX	7FFD0000
ESP	0012FFC0
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910200 ntdll.7C910200
EIP	004015A3 Halware_.004015A3

DOPO:

Address	Disassembly	Comment
00401577	PUSH EBP	
00401578	MOV EBP, ESP	
0040157A	PUSH -1	
0040157C	PUSH Halware_.004040C0	
00401581	PUSH Halware_.0040203C	SE handler installation
00401584	MOV EAX, DWORD PTR FS:[0]	
0040158C	PUSH EAX	
00401590	MOV DWORD PTR FS:[0], ESP	
00401594	SUB ESP, 10	
00401597	PUSH EBX	
00401598	PUSH ESI	
00401599	PUSH EDI	
0040159A	MOV DWORD PTR SS:[EBP-10], ESP	
0040159D	CALL DWORD PTR DS:[<KERNEL32.GetVersion	kernel32.GetVersion
004015A2	XOR EDX, EDX	
004015A3	MOV DL, AH	
004015A7	MOV DWORD PTR DS:[405204], EAX	

Register	Value
EAX	00200105
ECK	7FFD0000
EDI	00000000
EBX	7FFD0000
ESP	0012FFC0
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910200 ntdll.7C910200
EIP	004015A5 Halware_.004015A5



Nel dettaglio, l'istruzione esegue l'AND logico sui bit di ECX e del valore esadecimale FF. Prima di tutto, convertiamo entrambi i valori in formato binario e quindi eseguiamo l'operazione di AND logico tra i rispettivi bit.

Esadecimale	Binario
0A280105	0000 1010 0010 1000 0000 0001 0000 0101
FF	0000 0000 0000 0000 0000 0000 1111 1111

Eseguendo l'AND logico tra i bit uno ad uno 0000 0000 0000 0000 0000 0000 0000 0101

Che in esadecimale corrisponde a: 00000005

Così abbiamo il valore di ECX dopo l'istruzione AND ECX, FF



3° ESERCIZIO

Prima di procedere con l'istruzione di "step-into" dopo aver impostato il secondo breakpoint all'indirizzo di memoria 004015AF e avendo il valore del registro ECX iniziale come "0A280103", convertiamo questo valore in formato binario.

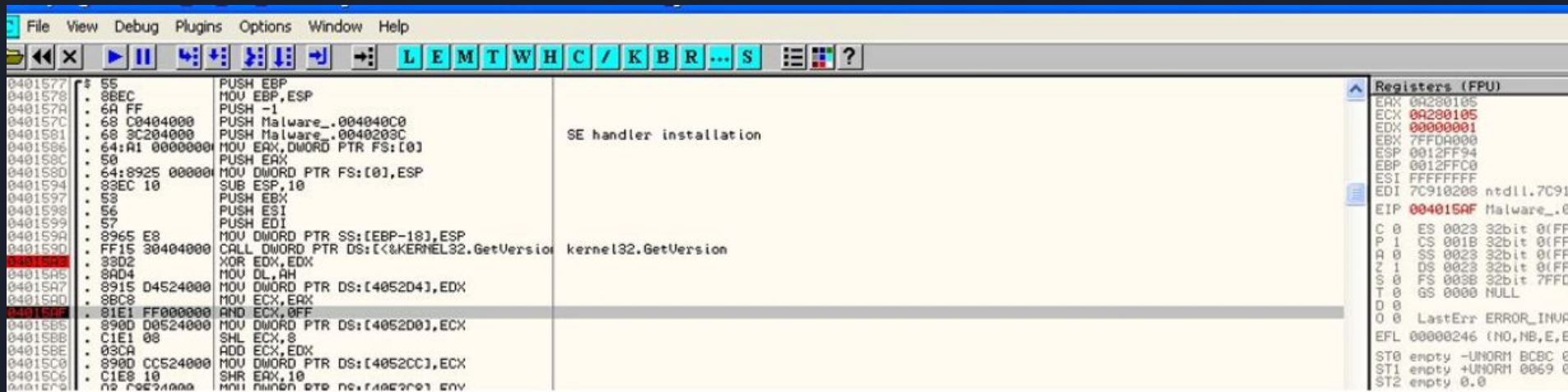
Il valore in esadecimale "0A280103" corrisponde al seguente valore binario: 00001010 00101000 00000001 00000011

Dopo aver eseguito l'istruzione di "step-into", il valore del registro ECX è stato modificato dall'esecuzione del programma. Per capire quale istruzione è stata eseguita, dobbiamo analizzare il codice macchina all'indirizzo di memoria 004015AF e il contesto del programma.

Dopo aver eseguito l'istruzione di "step-into" e raggiunto il secondo breakpoint, il nuovo valore del registro ECX sarà determinato dall'istruzione eseguita alla posizione di memoria 004015AF. Questa istruzione non è specificata nella tua richiesta, quindi non posso fornire una risposta precisa senza tale informazione.

Se fornisci l'istruzione esatta all'indirizzo di memoria 004015AF, posso aiutarti a determinare il nuovo valore di ECX e spiegare quale operazione è stata eseguita.

PRIMA



The screenshot shows a debugger window with the following components:

- Menu Bar:** File, View, Debug, Plugins, Options, Window, Help.
- Toolbar:** Navigation and execution controls, including a toolbar with letters L, E, M, T, W, H, C, /, K, B, R, ..., S.
- Assembly View:**

Address	Disassembly	Comment
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
004015A3	33D2	XOR EDX,EDX
004015A5	8AD4	MOV DL,AH
004015A7	8915 04524000	MOV DWORD PTR DS:[4052D4],EDX
004015AD	8BC8	MOV ECX,ERX
004015B0	81E1 FF000000	AND ECX,0FF
004015B5	890D 00524000	MOV DWORD PTR DS:[4052D0],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	02 C0524000	MOV EAX,DWORD PTR DS:[4052C0] EAX
- Registers (FPU):**

Register	Value
EAX	0A280105
ECX	0A280105
EDX	00000001
EBX	7FFD0000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.7C91
EIP	004015AF Malware_.0
C 0	ES 0023 32bit 0(F
P 1	CS 001B 32bit 0(F
A 0	SS 0023 32bit 0(F
Z 1	DS 0023 32bit 0(F
S 0	FS 003B 32bit 7FFD
T 0	GS 0000 NULL
D 0	
O 0	LastErrr ERROR_INVA
EFL	00000246 (NO,NB,E,E
ST0	empty -UNORM BCBC 0
ST1	empty +UNORM 0069 0
ST2	empty 0.0

DOPO

Dopo l'istruzione "step-into", il contenuto del registro ECX è stato alterato e ora risulta essere "00000005" a seguito dell'esecuzione dell'operazione logica AND con il valore "FF".

OllyDbg - Malware_U3_W3_L3.exe - [CPU - main thread, module Malware_]

File View Debug Plugins Options Window Help

LEMTWHC / KBR... S

```
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
00401590 64:9925 000000 MOV DWORD PTR FS:[0],ESP
00401594 83EC 10 SUB ESP,10
00401597 68 PUSH EAX
00401598 56 PUSH ESI
00401599 57 PUSH EDI
0040159A 8965 E8 MOV DWORD PTR SS:[EBP-18],ESP
0040159B FF15 30404000 CALL DWORD PTR DS:[<&KERNEL32.GetVersion
0040159D 3302 XOR EDX,EDX
004015A5 8B04 MOV ECX, [4052D04],EDX
004015A7 8B15 04524000 MOV ECX, EAX
004015AD 8BC8 MOV ECX, EAX
004015B2 81E1 FF000000 AND ECX, 0FF
004015B5 8900 00524000 MOV DWORD PTR DS:[4052D08],ECX
004015B8 C1E1 08 SHL ECX, 8
004015BE 03CA ADD ECX, EDX
004015C0 8900 CC524000 MOV DWORD PTR DS:[4052D0C],ECX
004015C6 C1E8 10 SHR EAX, 10
004015C9 A2 C8524000 MOV DWORD PTR DS:[4052D08],EAX
004015CF 6A 00 PUSH 0
004015D0 E8 33890000 CALL Malware_.00401F00
004015D5 59 POP ECX
004015D6 85C0 TEST EAX,EAX
004015D7 5E POP ESI
```

SE handler installation

kernel32.GetVersion

Registers (FPU)

```
EAX 0A200105
ECX 00000005
EDX 00000001
EBX 7FFD0000
ESP 0012FF94
EBP 0012FFC0
ESI FFFFFFFF
EDI 7C910200 ntdll.7C910200
EIP 004015B5 Malware_.004015B5
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 0018 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 0038 32bit 7FFDF000(FFF)
T 0 GS 0000 NULL
D 0
0 0 LastErr ERROR_INVALID_HANDLE (00000000)
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
```




CONSIDERAZIONE FINALE

Il funzionamento completo del malware potrebbe includere molte altre fasi e comportamenti, come il caricamento di componenti aggiuntivi, la comunicazione con server di comando e controllo remoto, l'esecuzione di azioni dannose come il furto di informazioni o l'infezione di altri file.