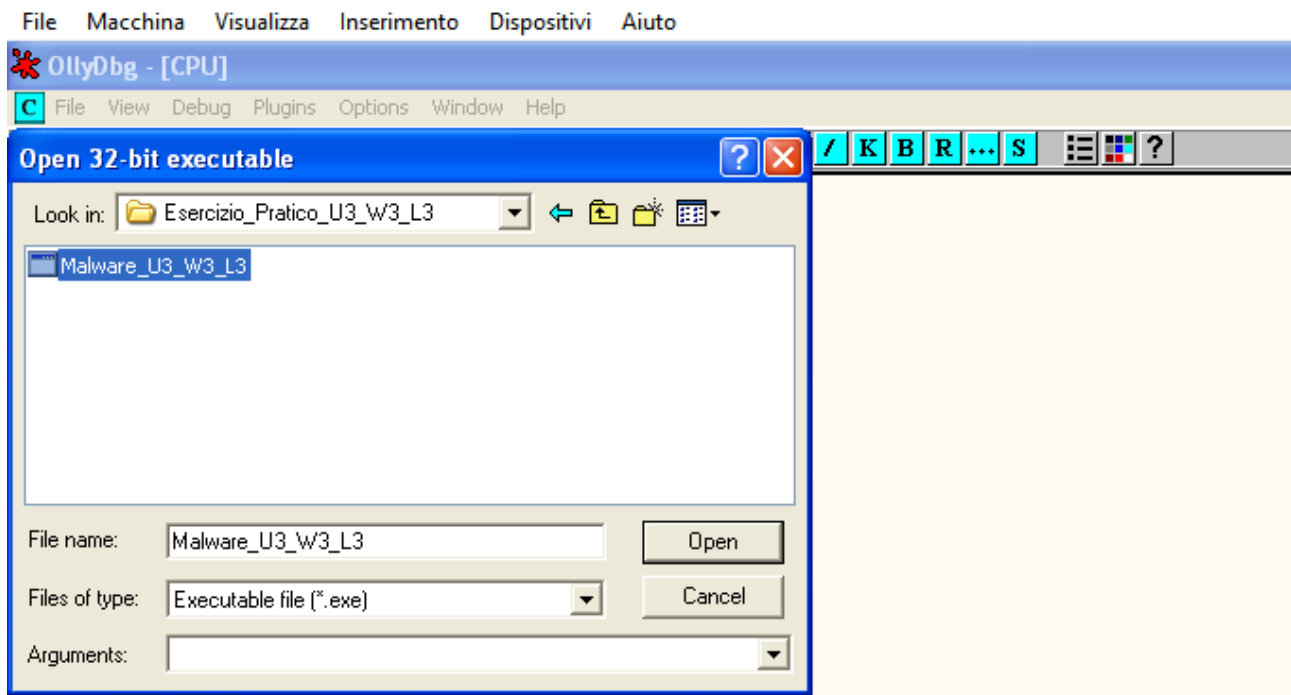


Traccia:

Fate riferimento al malware: **Malware_U3_W3_L3**, presente all'interno della cartella **Esercizio_Pratico_U3_W3_L3** sul desktop della macchina virtuale dedicata all'analisi dei malware. Rispondete ai seguenti quesiti utilizzando OlllyDBG.

- All'indirizzo 0040106E il Malware effettua una chiamata di funzione alla funzione «CreateProcess». Qual è il valore del parametro «CommandLine» che viene passato sullo stack? (1)
- Inserite un breakpoint software all'indirizzo 004015A3. Qual è il valore del registro EDX? (2) Eseguite a questo punto uno «step-into». Indicate qual è ora il valore del registro EDX (3) motivando la risposta (4). Che istruzione è stata eseguita? (5)
- Inserite un secondo breakpoint all'indirizzo di memoria 004015AF. Qual è il valore del registro ECX? (6) Eseguite un step-into. Qual è ora il valore di ECX? (7) Spiegate quale istruzione è stata eseguita (8).
- BONUS: spiegare a grandi linee il funzionamento del malware

Per iniziare apriamo il file eseguibile del malware con OlllyDBG per analizzare dinamicamente le funzioanlità del malware



1.

Come richiesto dalla prima task, all'indirizzo di memoria 0040106E notiamo che il valore del parametro "commandLine" passato allo stack è "cmd"

00401053	. 8D55 F0	LEA EDX,DWORD PTR SS:[EBP-10]	
00401056	. 52	PUSH EDX	
00401057	. 8D45 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	

```
pProcessInfo
pStartupInfo
CurrentDir = NULL
pEnvironment = NULL
CreationFlags = 0
InheritHandles = TRUE
pThreadSecurity = NULL
pProcessSecurity = NULL
CommandLine = "cmd"
ModuleFileName = NULL
CreateProcessA
```

2.

Posizionandoci sull'indirizzo 004015A3 con il tasto destro inseriamo l'indirizzo di memoria come breakpoint per il nostro debug e di conseguenza l'indirizzo di memoria si colorerà di rosso

CPU - main thread, module Malware_			
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	kernel32.GetVersion
004015A3	. 33D2	XOR EDX,EDX	
004015A5	. 8A04	MOV DL,AH	
004015A7	. 8915 D4524000	MOV DWORD PTR DS:[405204],EDX	

Andiamo a individuare il valore di EDX dal registro

Registers (FPU)	
EAX	0A280105
ECX	7FFDC000
EDX	00000A28
EBX	7FFDC000
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	
O 0	LastErr ERROR_INVALID_HANDLE (00000006)
EFL	00000206 (NO,NB,NE,A,NS,PE,GE,G)
ST0	empty -UNORM BDEC 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)

3.

Successivamente procedendo con lo step-into, andiamo a controllare se il valore del registro EDX sia cambiato o meno

```
Registers (FPU)
EAX 0A280105
ECX 7FFDC000
EDX 00000000 ←
EBX 7FFDC000
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 7FFDF000(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 BDEC 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,53 Mask 1 1 1 1 1 1
```

4. 5.

Il risultato è cambiato ed è diventato 0 per via dell'operatore logico XOR, il quale finchè avrà due valori uguali darà come risultato 0

6.

Successivamente procediamo a mettere un secondo breakpoint all'indirizzo di memoria 004015AF utilizzando lo stesso metodo di prima

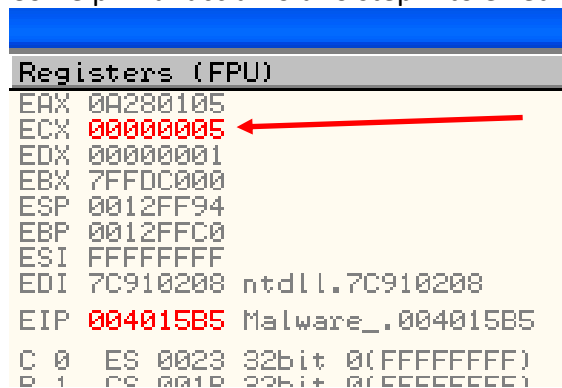
```
CPU - main thread, module Malware_
00401594 83EC 10 SUB ESP,10
00401597 53 PUSH EBX
00401598 56 PUSH ESI
00401599 57 PUSH EDI
0040159A 8965 E8 MOV DWORD PTR SS:[EBP-18],ESP
0040159D FF15 30404000 CALL DWORD PTR DS:[<&KERNEL32.GetVersion kernel32.GetVersion
004015A3 33D2 XOR EDX,EDX
004015A5 8AD4 MOV DL,AH
004015A7 8915 D4524000 MOV DWORD PTR DS:[4052D4],EDX
004015AD 8BC8 MOV ECX,EAX
004015AF 81E1 FF000000 AND ECX,0FF
004015B5 890D D0524000 MOV DWORD PTR DS:[4052D0],ECX
```

E vedendo dal registro stavolta il valore di ECX = 0A280105

```
Registers (FPU)
EAX 0A280105
ECX 0A280105 ←
EDX 00000001
EBX 7FFDC000
ESP 0012FF94
EBP 0012FFC0
ESI FFFFFFFF
EDI 7C910208 ntdll.7C910208
EIP 004015AF Malware_.004015AF
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 7FFDF000(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 BDEC 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,53 Mask 1 1 1 1 1 1
```

7.

Come prima facciamo uno step-into e vediamo come cambia il valore del registro ECX = 00000005



```
Registers (FPU)
EAX 0A280105
ECX 00000005
EDX 00000001
EBX 7FFDC000
ESP 0012FF94
EBP 0012FFC0
ESI FFFFFFFF
EDI 7C910208 ntdll.7C910208
EIP 004015B5 Malware_.004015B5
C 0 ES 0023 32bit 0(FFFFFFFF)
D 1 CS 0010 32bit 0(FFFFFFFF)
```

8.

Eseguendo l'operatore logico AND tra il valore di ECX e OFF verrà fuori il nuovo valore del registro ECX come si vede sopra