

Cyber Security & Ethical Hacking

GIORNO 3 –MALWARE ANALYSIS

Alejandro Cristino
S11-L3

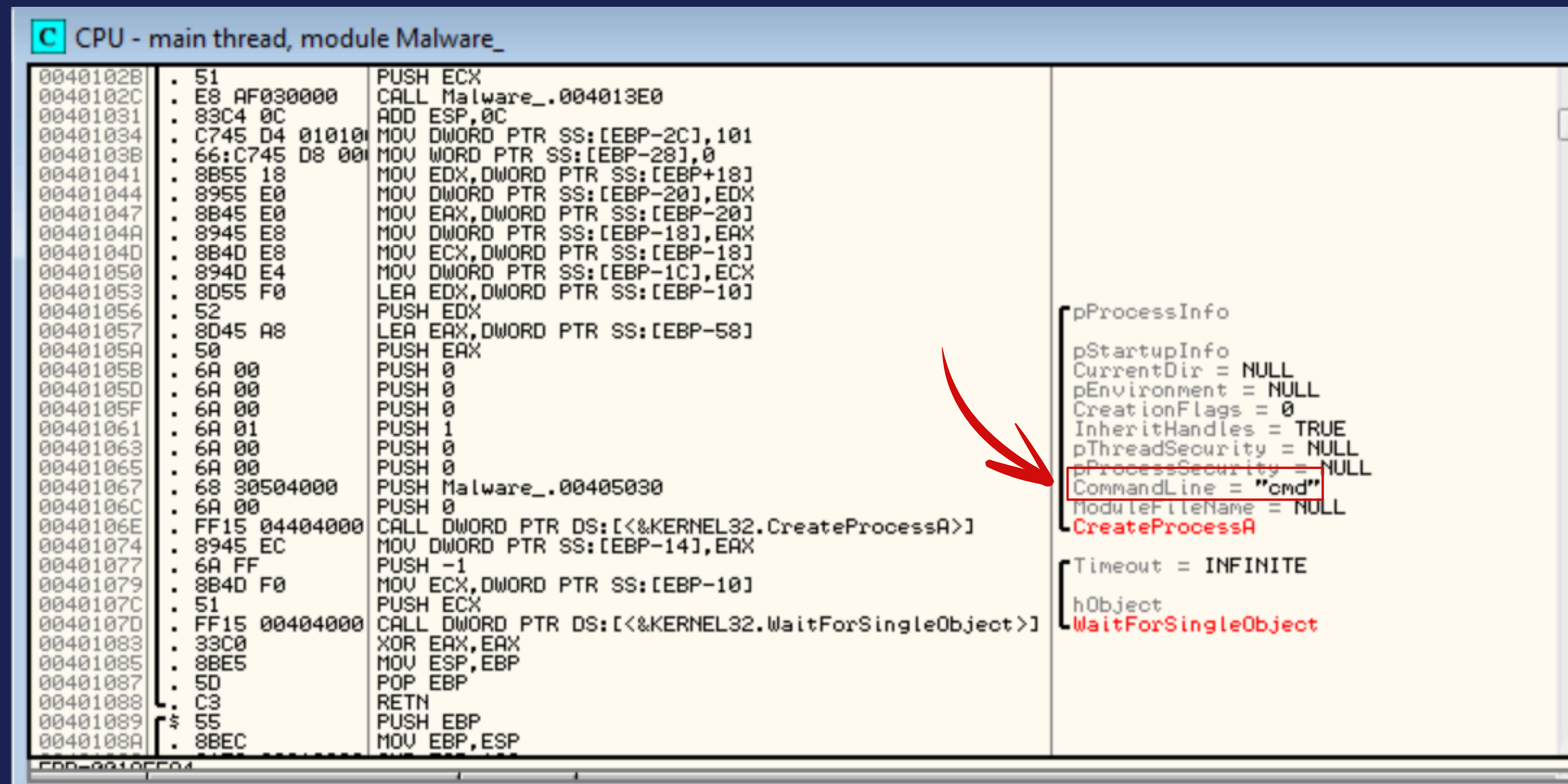
TRACCIA

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 OllyDBG.

- 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

1. All'indirizzo 0040106E il Malwareeffettua una chiamata di funzione alla funzione «CreateProcess». Qual è il valore del parametro «CommandLine» che viene passato sullo stack?

Il valore del parametro CommandLine che passa sullo stack è "cmd"



```
CPU - main thread, module Malware_  
0040102B . 51 PUSH ECX  
0040102C . E8 AF030000 CALL Malware_.004013E0  
00401031 . 83C4 0C ADD ESP,0C  
00401034 . C745 D4 010101 MOV DWORD PTR SS:[EBP-2C],101  
0040103B . 66:C745 D8 0000 MOV WORD PTR SS:[EBP-28],0  
00401041 . 8B55 18 MOV EDX,DWORD PTR SS:[EBP+18]  
00401044 . 8955 E0 MOV DWORD PTR SS:[EBP-20],EDX  
00401047 . 8B45 E0 MOV EAX,DWORD PTR SS:[EBP-20]  
0040104A . 8945 E8 MOV DWORD PTR SS:[EBP-18],EAX  
0040104D . 8B4D E8 MOV ECX,DWORD PTR SS:[EBP-18]  
00401050 . 894D E4 MOV DWORD PTR SS:[EBP-1C],ECX  
00401053 . 8D55 F0 LEA EDI,DWORD PTR SS:[EBP-10]  
00401056 . 52 PUSH EDI  
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]  
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 04404000 CALL DWORD PTR DS:[&KERNEL32.WaitForSingleObject]  
00401083 . 33C0 XOR EAX,EAX  
00401085 . 8BE5 MOV ESP,EBP  
00401087 . 5D POP EBP  
00401088 . C3 RETN  
00401089 . 55 PUSH EBP  
0040108A . 8BEC MOV EBP,ESP
```

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

2. Inserite un breakpoint software all'indirizzo 004015A3. Qual è il valore del registro EDX? 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?

Inserendo un breakpoint all'indirizzo 004015A3, identifichiamo il valore iniziale del registro EDX come 00001DB1

The screenshot shows a debugger window titled "CPU - main thread, module Malware_". The instruction list on the left shows the execution of `XOR EDX, EDX` at address 004015A3, which is highlighted with a red box. A red arrow points from this instruction to the "Registers (FPU)" window on the right. In the registers window, the EDX register is highlighted with a red box and shows the value 00001DB1.

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, EAX

Registers (FPU)

Register	Value
EAX	1DB10106
ECX	7EFDE000
EDX	00001DB1
EBX	7EFDE000
ESP	0018FF5C
EBP	0018FF88
ESI	00000000
EDI	00000000
EIP	004015A3 Malware_.004015A3

Eseguiamo a questo punto uno step-into, dove possiamo subito notare che il valore del registro EDX sia cambiato, in questo caso, dopo un'operazione XOR, il valore del registro viene settato a 0.

The screenshot shows the same debugger window after a step-into operation. The instruction list now shows the execution of `MOV DL, AH` at address 004015A5, which is highlighted with a red box. A red arrow points from this instruction to the "Registers (FPU)" window on the right. In the registers window, the EDX register is highlighted with a red box and shows the value 00000000.

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, EAX
004015AF	81E1 FF000000	AND ECX, 0FF
004015B5	890D 00524000	MOV DWORD PTR DS:[4052D0], ECX
004015B8	C1E1 08	SHL ECX, 8

Registers (FPU)

Register	Value
EAX	1DB10106
ECX	7EFDE000
EDX	00000000
EBX	7EFDE000
ESP	0018FF5C
EBP	0018FF88
ESI	00000000
EDI	00000000
EIP	004015A5 Malware_.004015A5

3. Inserite un secondo breakpoint all'indirizzo di memoria 004015AF. Qual è il valore del registro ECX? Eseguite un step-into. Qual è ora il valore di ECX? Spiegate quale istruzione è stata eseguita.

Inserendo un breakpoint all'indirizzo 004015AF, identifichiamo il valore iniziale del registro ECX come **1DB10106**

The screenshot shows a debugger window titled "CPU - main thread, module Malware_". The instruction list on the left includes:
0040158C: 50 PUSH EAX
0040158D: 64:8925 00000 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 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
004015B8: C1E1 08 SHL ECX,8
004015BE: 83C9 ADD ECX,ECX
The "Registers (FPU)" window on the right shows the initial value of ECX as **1DB10106**, which is highlighted with a red box. A red arrow points from the instruction list to the register window.

Eseguendo uno step-into, il valore di ECX ora è **00000006**

The screenshot shows the same debugger window after a step-into operation. The instruction list is the same, but the current instruction pointer (EIP) is now 004015B5, and the instruction being executed is "MOV DWORD PTR DS:[4052D0],ECX". The "Registers (FPU)" window on the right shows the updated value of ECX as **00000006**, which is highlighted with a red box. A red arrow points from the instruction list to the register window.

L'istruzione esegue l'AND logico sui bit di ECX e del valore esadecimale OFF, poi si portano entrambi i valori in binario e si esegue l'AND logico tra i bit uno ad uno 0000 0000 0000 0000 0000 0000 0000 0110 che in esadecimale corrisponde a **00000006**

Cyber Security & Ethical Hacking

GRAZIE.