

# S11 L3

Report Esercizio Malware Analysis

GiuliaSalani

## INDICE

<b>TRACCIA</b> .....	2
<b>SVOLGIMENTO</b> .....	3
<b>DEFINIZIONE: OLLY DBG</b> .....	3
<b>PREPARAZIONE</b> .....	3
<b>QUESITO 1</b> .....	5
<b>QUESITO 2</b> .....	6
<b>QUESITO 3</b> .....	8
<b>QUESITO 4 - BONUS</b> .....	9

## 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 uno 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

## SVOLGIMENTO

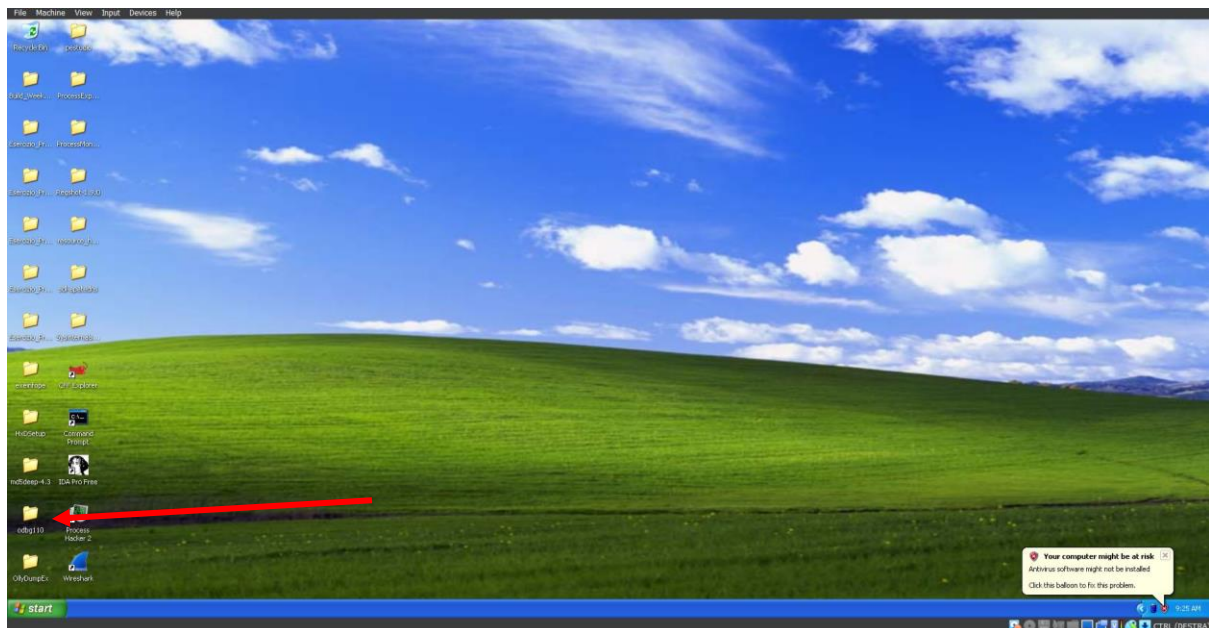
### DEFINIZIONE: OLLY DBG



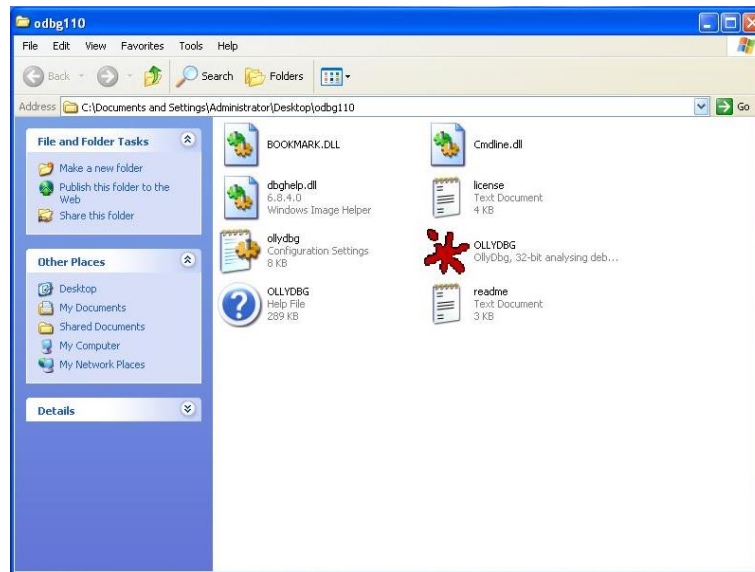
OllyDbg è un **debugger per analisi e reverse engineering di software**. Si tratta di un'applicazione software utilizzata principalmente nel campo della sicurezza informatica e dello sviluppo software. OllyDbg consente agli sviluppatori e agli esperti di sicurezza di esaminare il codice eseguibile di un programma, eseguire il debug in modalità interattiva, analizzare la memoria e monitorare il flusso di esecuzione del programma. Questo strumento è particolarmente apprezzato per la sua interfaccia utente intuitiva e le potenti funzionalità di analisi binaria, rendendolo una risorsa preziosa per la comprensione approfondita del comportamento interno di un'applicazione.

### PREPARAZIONE

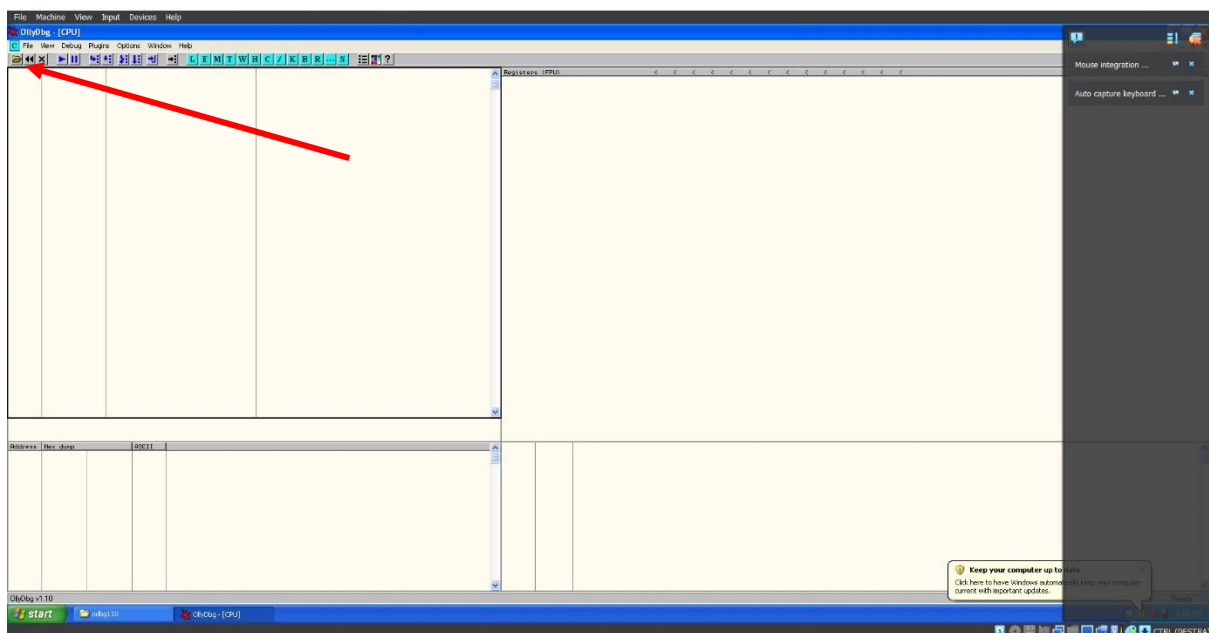
Come prima cosa, prepariamo il software e carichiamo il malware oggetto dell'analisi odierna. Apriamo la macchina virtuale e individuiamo la cartella che contiene OllyDBG sul desktop:



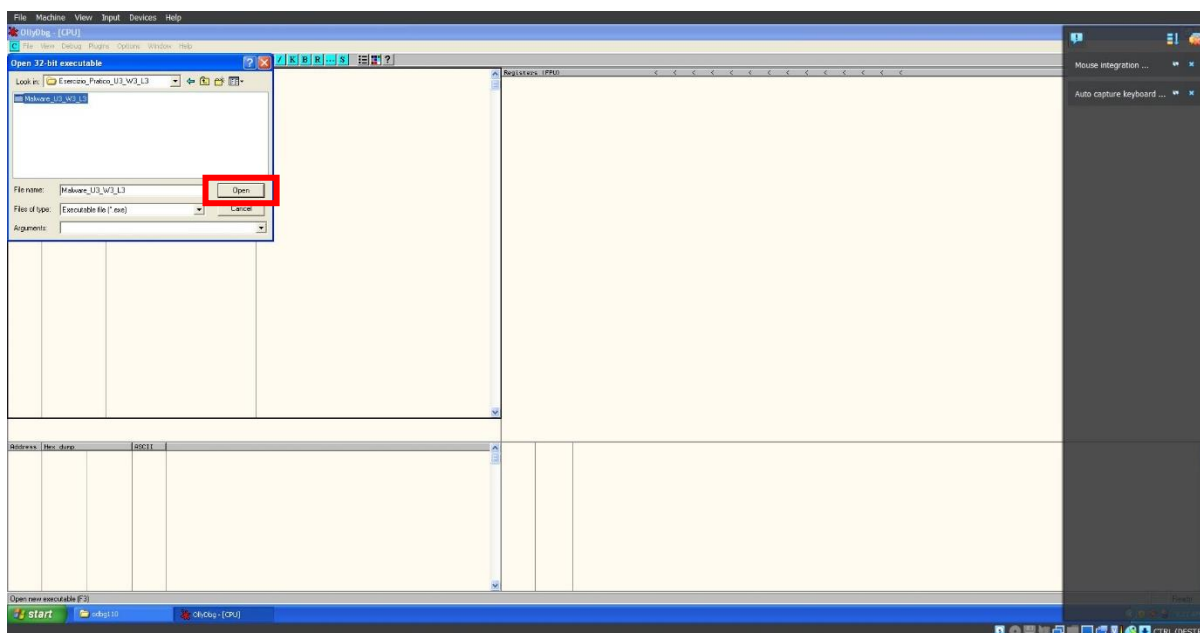
Individuiamo l'eseguibile e con doppio click lo facciamo partire:



Clicchiamo l'icona a forma di cartella in alto a sinistra:



Selezioniamo il file da analizzare e clicchiamo su open:

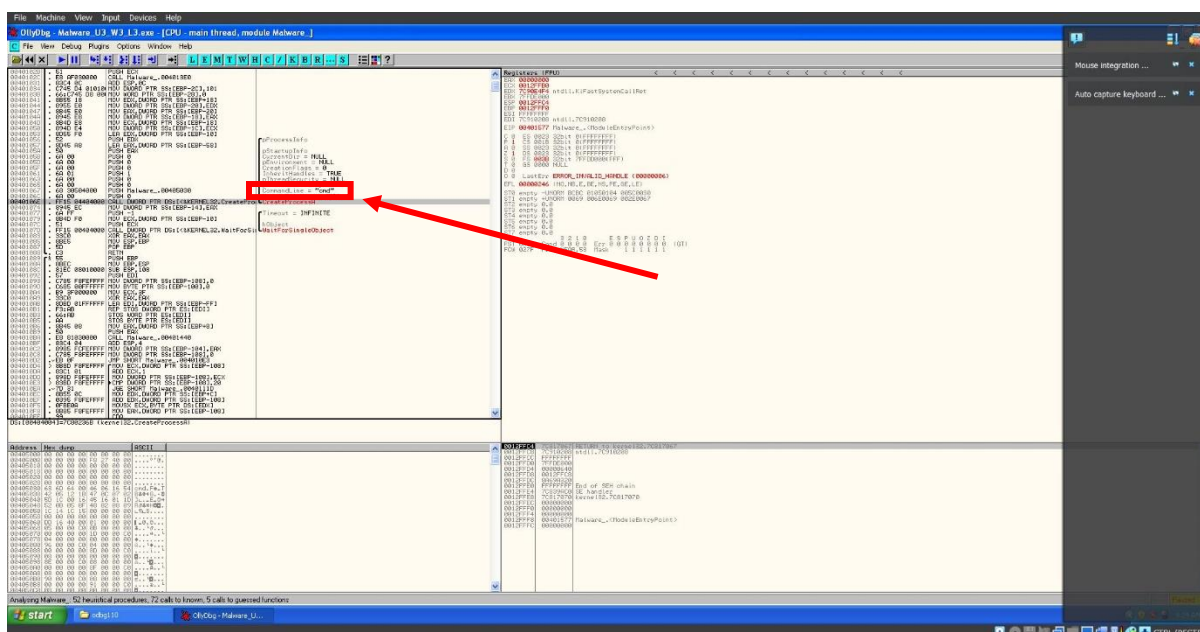


Ora possiamo affrontare le consegne.

## QUESITO 1

**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)**

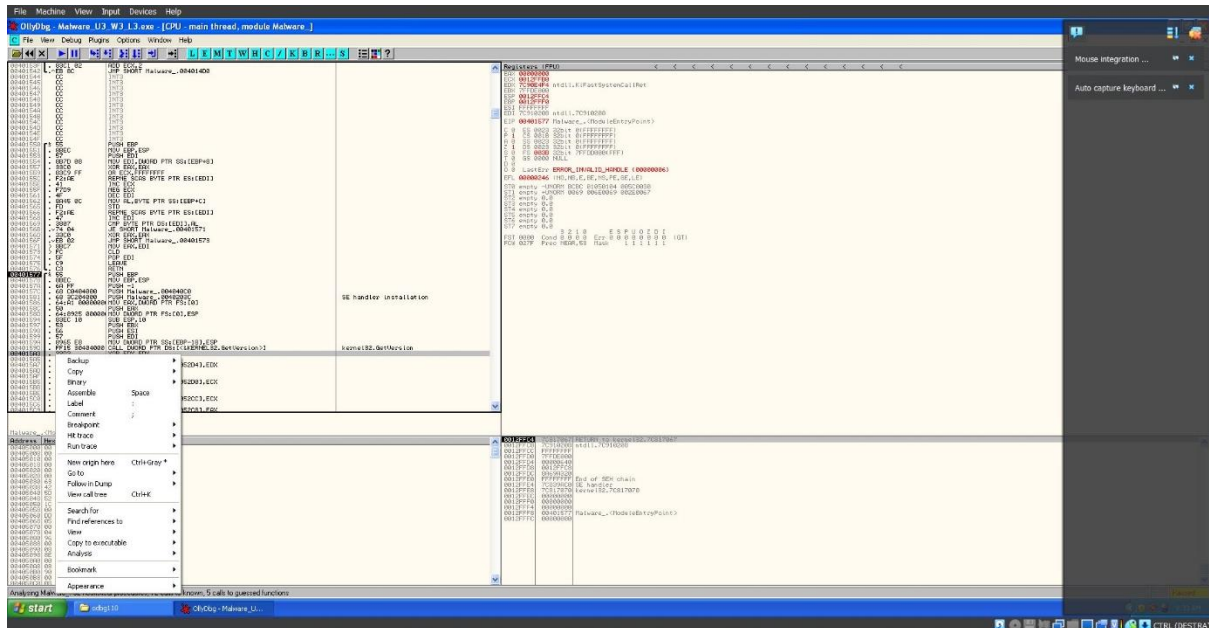
Concentriamoci sul riquadro in alto a sinistra. Scorriamo la lista degli indirizzi di memoria a sinistra fino a incontrare l'indirizzo indicato in consegna. Il parametro di CommandLine è indicato nella quarta colonna e corrisponde a "cmd":



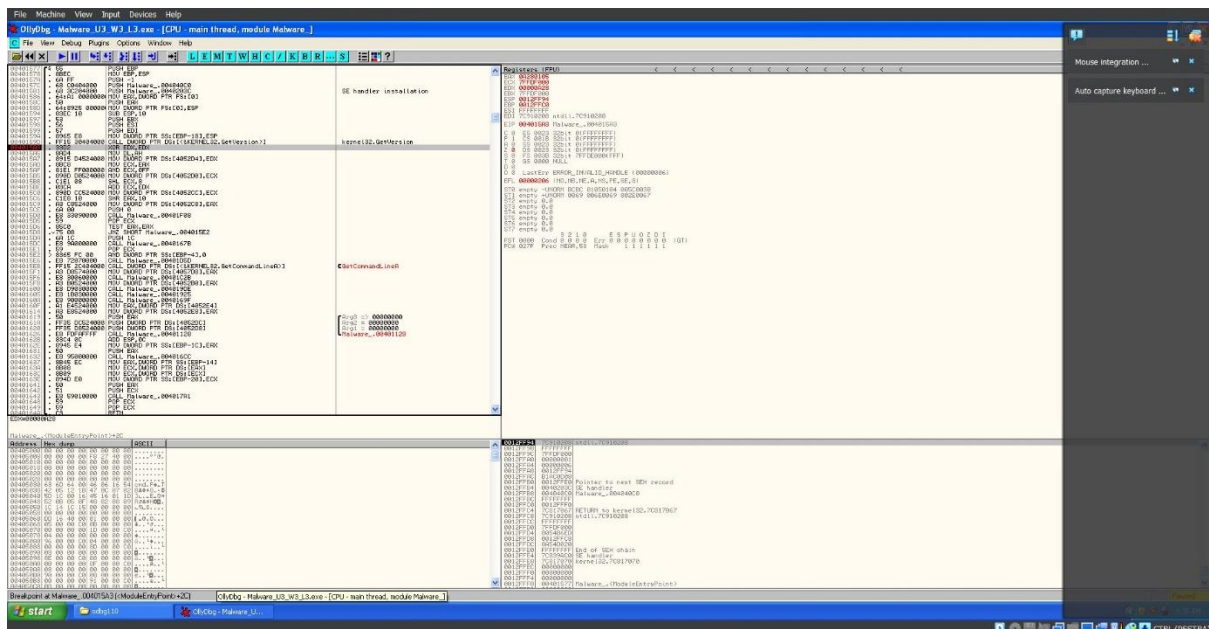
## QUESITO 2

**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)**

Scorriamo la colonna a sinistra fino a trovare l'indirizzo indicato in questa consegna. Per inserire un breakpoint clicchiamo con il tasto destro sulla riga e alla voce breakpoint selezioniamo il toggle (che sarebbe il software breakpoint):



Il valore del registro EDX lo vediamo nel quadrante di fianco a quello utilizzato finora:



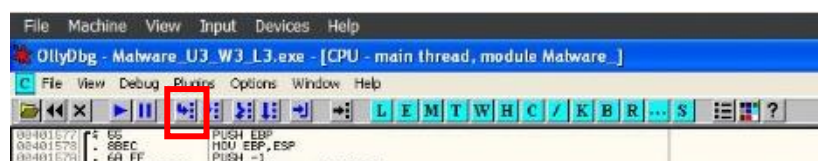
Il valore del registro di EDX è 00000A28:

```

Registers (FPU)
EAX 00280105
ECX 77FD0000
EDX 00000A28
EBX 77FD0000
ESP 0012CF94
EBP 0012CF00
ESI FFFFFFFF
EDI 7C910200 ntdll!_7C910200
EIP 00401503 NtLdr!_00401503
C 9 ES 0023 32bit 0FFFFFFFF
P 1 CS 001B 32bit 0FFFFFFFF
D 0 SS 0023 32bit 0FFFFFFFF
I 0 DS 0023 32bit 0FFFFFFFF
T 0 FS 0023 32bit 77FD0000
D 0 SS 0000 NULL
D 0 LastErr ERROR_INVALID_HANDLE (00000006)
EFL 00000206 IN0,HE,A,HS,FE,SE,SI
ST0 empty -UNORM BCBC 01050104 00C00000
ST1 empty -UNORM 0069 00E00069 00C00067
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
FPU 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,S3 Task 1 1 1 1 1 1

```

Nella barra in alto selezioniamo questo tasto per effettuare lo step into:



Il nuovo valore di EDX è zero (00000000):

```

Registers (FPU)
EAX 00280105
ECX 77FD0000
EDX 00000000
EBX 77FD0000
ESP 0012CF94
EBP 0012CF00
ESI FFFFFFFF
EDI 7C910200 ntdll!_7C910200
EIP 00401506 NtLdr!_00401506
C 9 ES 0023 32bit 0FFFFFFFF
P 1 CS 001B 32bit 0FFFFFFFF
D 0 SS 0023 32bit 0FFFFFFFF
I 0 DS 0023 32bit 0FFFFFFFF
T 0 FS 0023 32bit 77FD0000
D 0 SS 0000 NULL
D 0 LastErr ERROR_INVALID_HANDLE (00000006)
EFL 00000206 IN0,HE,E,FE,HS,FE,SE,LEI
ST0 empty -UNORM BCBC 01050104 00C00000
ST1 empty -UNORM 0069 00E00069 00C00067
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
FPU 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,S3 Task 1 1 1 1 1 1

```

L'istruzione eseguita è stata XOR EDX, EDX il cui scopo è proprio azzerare il registro:

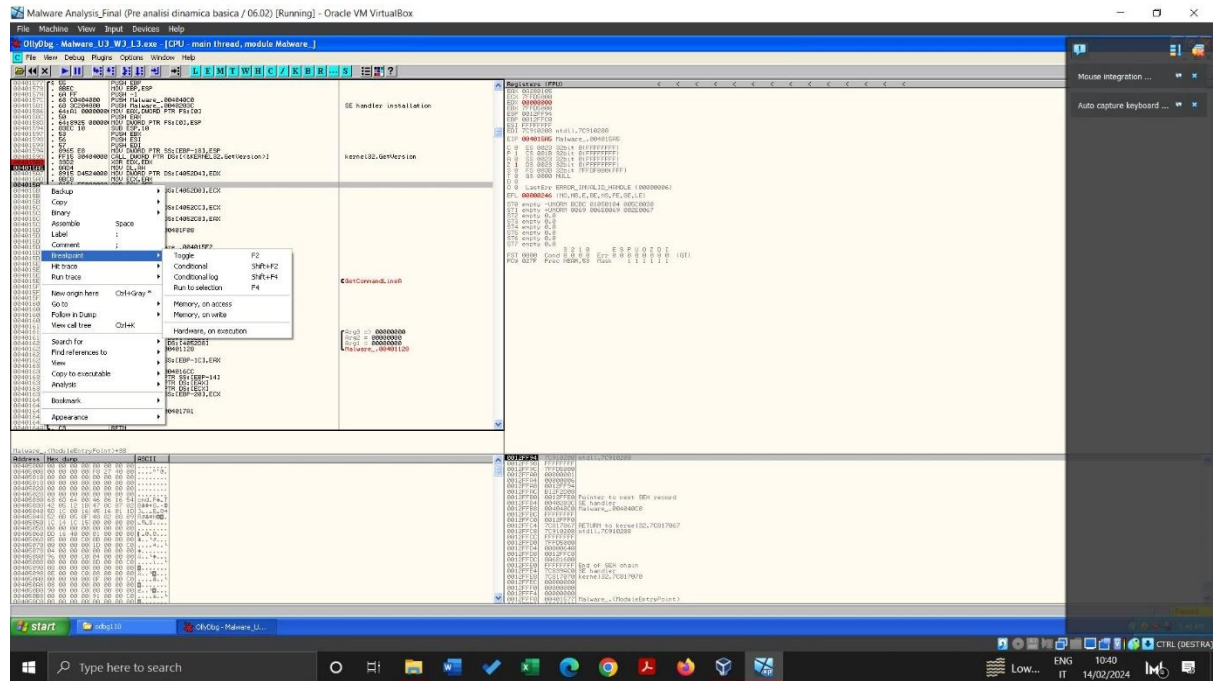




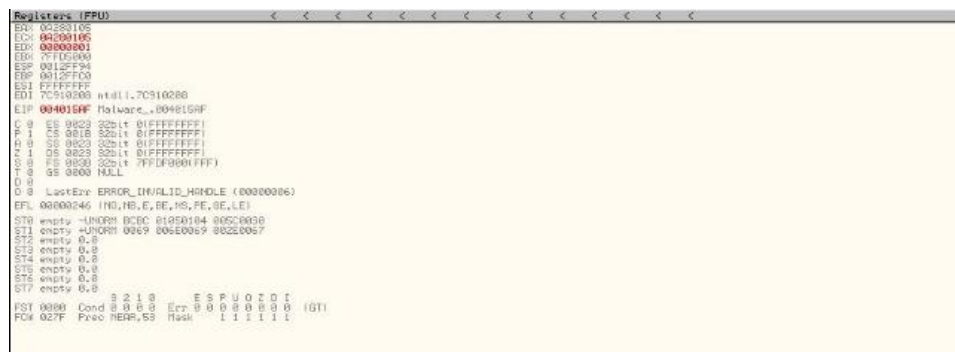
## QUESITO 3

**Inserite un secondo breakpoint all'indirizzo di memoria 004015AF. Qual è il valore del registro ECX? (6) Eseguite uno step-into. Qual è ora il valore di ECX? (7) Spiegate quale istruzione è stata eseguita (8).**

Individuiamo il nuovo indirizzo e inseriamo un toggle breakpoint come fatto precedentemente:



Il valore di ECX è pari a 0A280105:



Dopo l'operazione diventa 00000005:



L'istruzione eseguita è AND ECX, 0FF.

Il valore "0FF" sembra essere in esadecimale; l'istruzione AND ECX, 0FF sta effettuando un'operazione AND a livello bit tra il registro ECX e il valore esadecimale "0FF".

"0FF" in binario sarebbe "000011111111" (8 bit con tutti i bit da 1). Quindi, l'operazione di AND con "0FF" ha l'effetto di mantenere solo i primi 8 bit meno significativi di ECX, azzerando tutti gli altri bit.

Considerato il valore iniziale di ECX "0A280105" (ad esempio, in binario: "0000101000101000000100000000101"), l'operazione di AND con "0FF" produce il risultato "00000000000000000000000000101" (che è 5 in decimale).

## QUESITO 4 - BONUS

**Spiegare a grandi linee il funzionamento del malware**

Scorrendo il flusso del programma, è possibile distinguere la presenza di funzioni come chiamate di rete, manipolazione di stringhe, gestione della memoria e funzioni legate all'interfaccia utente.

In questo punto, ad esempio, il malware crea un processo:

The screenshot shows the OllyDbg interface with the assembly code of the malware's main thread. The code is in x86 assembly and includes instructions for stack manipulation, pushing/popping registers, and calling the Windows API function `CreateProcessA`. The registers window on the right shows the current state of the CPU registers, including EAX, ECX, EDI, and ESI. The registers window also displays the current instruction pointer (EIP) and the current instruction (C0: 00401585: Malware\_.00401585).

Qui crea un socket per connettersi con un server remoto:

```

00401253 > 0F04 F9010000 JE Malware_.00401304
00401259 > 8085 68FEFFFF LEA EAX,DWORD PTR SS:[EBP-198]
0040125F > 50 PUSH EAX
00401260 > 68 02000000 PUSH 200
00401265 > FF15 9C404000 CALL DWORD PTR DS:[&WS2_32.#115]
0040126B > 8985 4CFEFFFF MOV DWORD PTR SS:[EBP-184],EAX
00401271 > 83BD 4CFEFFFF CMP DWORD PTR SS:[EBP-184],0
00401278 > 74 0A JE SHORT Malware_.00401284
0040127F > B8 01000000 MOV EAX,1
00401284 > E9 52010000 JMP Malware_.00401306
0040128A > 6A 00 PUSH 0
0040128B > 6A 00 PUSH 0
0040128D > 6A 00 PUSH 0
0040128E > 6A 00 PUSH 0
0040128F > 6A 00 PUSH 0
00401290 > FF15 A0404000 CALL DWORD PTR DS:[&WS2_32.WSASocketA]
00401296 > 8985 FCFCFFFF MOV DWORD PTR SS:[EBP-304],EAX
0040129C > 83BD FCFCFFFF CMP DWORD PTR SS:[EBP-304],-1
004012A3 > 75 0A JNZ SHORT Malware_.004012AF
004012A5 > B8 01000000 MOV EAX,1
004012AA > E9 27010000 JMP Malware_.00401306
004012AF > 808D 10FEFFFF LEA ECX,DWORD PTR SS:[EBP-1F0]
004012B5 > 51 PUSH ECX
004012B6 > 8D4D 50FEFFFF LEA EDI,DWORD PTR SS:[EBP-1B0]
004012B8 > 52 PUSH EDI
004012BA > E8 C7DFFFFF CALL Malware_.00401089
004012BC > 8945 F8 MOV DWORD PTR SS:[EBP-8],EAX
004012BD > 8D45 F8 MOV ECX,DWORD PTR SS:[EBP-8]
004012BE > 58 PUSH EAX
004012BF > FF15 A4404000 CALL DWORD PTR DS:[&WS2_32.#52]
004012C0 > 8985 44FEFFFF MOV DWORD PTR SS:[EBP-1BC],EAX
004012C2 > 83BD 44FEFFFF CMP DWORD PTR SS:[EBP-1BC],0
004012C9 > 75 23 JNZ SHORT Malware_.0040137A
004012E1 > 8B8D FCFCFFFF MOV ECX,DWORD PTR SS:[EBP-304]
004012E7 > 51 PUSH ECX
004012E8 > FF15 A0404000 CALL DWORD PTR DS:[&WS2_32.#3]
004012ED > 8985 AC404000 MOV DWORD PTR DS:[&WS2_32.#116]
004012F4 > 68 30750000 PUSH 7530
004012F9 > FF15 00404000 CALL DWORD PTR DS:[&KERNEL32.Sleep]
004012FF > E9 40FEFFFF JMP Malware_.0040124C
00401304 > 8B45 F8 MOV EDI,DWORD PTR SS:[EBP-8]
00401305 > 8B42 0C MOV ECX,DWORD PTR DS:[EAX+C]
00401306 > 8B11 MOV EDI,DWORD PTR DS:[ECX]
00401307 > 8B11 MOV EDI,DWORD PTR DS:[ECX]
00401311 > 8985 38FEFFFF MOV DWORD PTR SS:[EBP-1C0],EDI
00401317 > 8B42 0C MOV ECX,DWORD PTR DS:[EAX+C]
0040131C > FF15 00404000 CALL DWORD PTR DS:[&WS2_32.#9]
00401322 > 66:8985 36FEFF MOV WORD PTR SS:[EBP-1CA],AX
00401323 > 66:C785 34FEFF MOV WORD PTR SS:[EBP-1CC],2
00401324 > 6A 10 PUSH 10
00401325 > 8D85 34FEFFFF LEA EAX,DWORD PTR SS:[EBP-1CC]
00401326 > 50 PUSH EAX
00401327 > 8B8D FCFCFFFF MOV ECX,DWORD PTR SS:[EBP-304]
00401328 > 51 PUSH ECX
00401329 > FF15 B4404000 CALL DWORD PTR DS:[&WS2_32.#4]
0040132A > 8985 4CFEFFFF MOV DWORD PTR SS:[EBP-1B4],EAX
0040132B > 83BD 4CFEFFFF CMP DWORD PTR SS:[EBP-1B4],-1
0040132C > 75 23 JNZ SHORT Malware_.0040137A
0040132D > 8B85 FCFCFFFF MOV ECX,DWORD PTR SS:[EBP-304]

```

Registers (FPU)

```

EAX 00200105
ECX 00000005
EDX 00000001
EBX 77FD5000
ESP 0012FF94
EBP 0012FFC0
ESI FFFFFFFF
EDI 7C910208 ntdll.7C910208
EIP 00401585 Malware_.00401
C 0 ES 0023 32bit 0FFFFFFFF
P 1 CS 0018 32bit 0FFFFFFFF
A 0 SS 0023 32bit 0FFFFFFFF
Z 0 DS 0023 32bit 0FFFFFFFF
S 0 FS 0038 32bit 77FD5000
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_INVALID
EFL 00000206 (NO,NO,NE,A,NS
ST0 empty -UNORM BCBC 0105
ST1 empty +UNORM 0069 006E
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
FCW 027F Prec NEAR,53 Mas

```

Flags = 0  
Group = 0  
pWSAProtocol = NULL  
Protocol = IPPROTO\_TCP  
Type = SOCK\_STREAM  
Family = AF\_INET  
WSASocketA

Arg2  
Arg1  
Malware\_.00401089

Name  
gethostbyname

Socket  
closesocket  
WSACleanup  
Timeout = 30000, ns  
Sleep

NetShort = 270F  
ntohs

AddrLen = 10 (16.)  
pSockAddr  
Socket  
connect

Invece, l'uso delle funzioni presenti nello screenshot di seguito potrebbe indicare un tentativo di creare o manipolare elementi dell'interfaccia utente, come visualizzare messaggi pop-up ingannevoli o fuorvianti, ad esempio falsi messaggi di errore o notifiche per ingannare gli utenti affinché compiano determinate azioni o per creare una distrazione:

```

00403359 > BA 00000000 MOV EDI,00000000
0040336E > 8BCB MOV ECX,EBX
00403390 > D3EA SHR EDI,CL
00403392 > F202 F202 AND EDI,2
00403394 > 2150 08 AND DWORD PTR DS:[EAX+8],EDI
00403397 > 8BC3 MOV EAX,EBX
00403399 > 5E POP ESI
0040339A > 5E POP ESI
0040339B > 5B POP EBX
0040339C > C9 LEAVE
0040339D > C9 LEAVE
0040339E > 53 PUSH EBX
0040339F > 330B XOR EBX,EBX
004033A1 > 91D 1C544000 CMP DWORD PTR DS:[40541C],EBX
004033A7 > 56 PUSH ESI
004033A8 > 57 PUSH EDI
004033A9 > 75 42 JNZ SHORT Malware_.004033ED
004033AB > 68 EC434000 PUSH Malware_.004043EC
004033AD > FF15 20404000 CALL DWORD PTR DS:[&KERNEL32.LoadLibraryA]
004033B6 > 8BF8 MOV EDI,EAX
004033B7 > 30F0 CIP EDI,EBX
004033BA > 74 67 JE SHORT Malware_.00403423
004033BC > 8B35 E4404000 MOV ESI,DWORD PTR DS:[&KERNEL32.GetProcAddress]
004033C2 > 68 E0434000 PUSH Malware_.004043E0
004033C7 > 51 PUSH EDI
004033C8 > FFD6 CALL ESI
004033C9 > 85C0 TEST EAX,EAX
004033CA > 85C0 TEST EAX,EAX
004033CB > A3 1C544000 MOV DWORD PTR DS:[40541C],EAX
004033D1 > 74 50 JE SHORT Malware_.00403423
004033D3 > 68 D0434000 PUSH Malware_.004043D0
004033D8 > 57 PUSH EDI
004033D9 > FFD6 CALL ESI
004033DB > 68 BC434000 PUSH Malware_.004043BC
004033DE > 57 PUSH EDI
004033E1 > A3 20544000 MOV DWORD PTR DS:[405420],EAX
004033E6 > FFD6 CALL ESI
004033E8 > A3 24544000 MOV DWORD PTR DS:[405424],EAX
004033ED > A3 20544000 MOV DWORD PTR DS:[405420],EAX
004033F2 > 85C0 TEST EAX,EAX
004033F4 > 74 16 JE SHORT Malware_.0040340C
004033F6 > FFD6 CALL EAX
004033F8 > 8B08 MOV EBX,EAX
004033FA > 85D0 TEST EBX,EBX
004033FB > 74 0E JE SHORT Malware_.0040340C
004033FC > A3 24544000 MOV ECX,DWORD PTR DS:[405424]
004033FD > 85C0 TEST EAX,EAX
004033FE > 74 05 JE SHORT Malware_.0040340C
00403400 > 53 PUSH EAX
00403401 > FFD0 CALL EAX
00403402 > 8B08 MOV EBX,EAX
00403403 > 57 PUSH EDI
00403404 > FF74 24 18 PUSH DWORD PTR SS:[ESP+18]
00403405 > FF74 24 18 PUSH DWORD PTR SS:[ESP+18]
00403406 > FF74 24 18 PUSH DWORD PTR SS:[ESP+18]
00403407 > FF74 24 18 PUSH DWORD PTR SS:[ESP+18]
00403408 > FF15 1C544000 CALL DWORD PTR DS:[40541C]
00403409 > 5F POP EDI
0040340A > 5E POP ESI
0040340B > 5B POP EBX
0040340C > C3 RETN
0040340D > 33C0 XOR EAX,EAX
0040340E > EB F8 JMP SHORT Malware_.0040341F

```

Registers (FPU)

```

EAX 00200105
ECX 00000005
EDX 00000001
EBX 77FD5000
ESP 0012FF94
EBP 0012FFC0
ESI FFFFFFFF
EDI 7C910208 ntdll.7C910208
EIP 00401585 Malware_.00401
C 0 ES 0023 32bit 0FFFFFFFF
P 1 CS 0018 32bit 0FFFFFFFF
A 0 SS 0023 32bit 0FFFFFFFF
Z 0 DS 0023 32bit 0FFFFFFFF
S 0 FS 0038 32bit 77FD5000
T 0 GS 0000 NULL
D 0
O 0 LastErr ERROR_INVALID
EFL 00000206 (NO,NO,NE,A,NS
ST0 empty -UNORM BCBC 0105
ST1 empty +UNORM 0069 006E
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
FCW 027F Prec NEAR,53 Mas

```

FileName = "user32.dll"  
LoadLibraryA

kernel32.GetProcAddress  
ProcName0rOrdinal = "MessageBoxA"  
hModule  
GetProcAddress

ProcName0rOrdinal = "GetActiveWindow"  
GetProcAddress  
ProcName0rOrdinal = "GetLastActivePopup"  
hModule  
GetProcAddress

Tutto questo potrebbe indicare un malware multifunzionale in grado di eseguire varie attività. Inoltre, l'uso di tecniche di offuscamento, crittografia o tecniche anti-analisi potrebbe suggerire un tentativo di eludere la rilevazione.

Convertendo la firma in hash con MD5DEEP e cercandola con l'aiuto di Virus Total, quest'ultimo suggerisce che si possa trattare di un Trojan.