

OLLYDBG

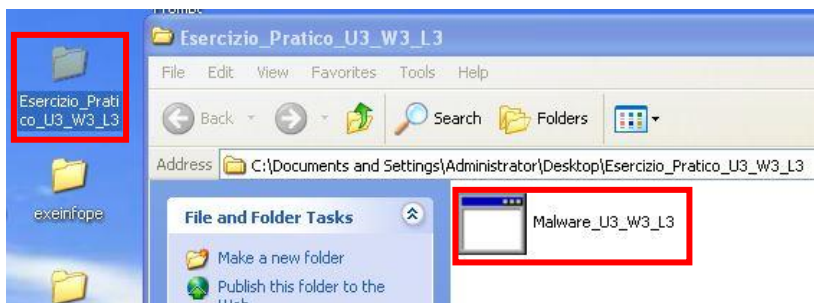
TASK

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**.

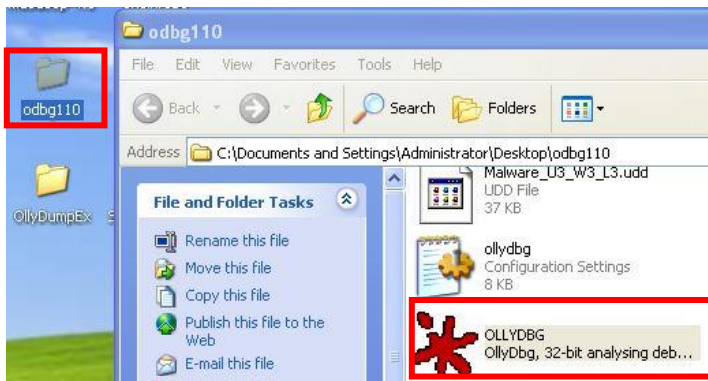
- All'indirizzo 0040106E il Malware effettua una chiamata di funzione alla funzione «**CreateProcess**». Qual è il valore del parametro «**CommandLine**» che viene passato sullo **stack**? (1)
- Inserite un **breakpoint** software all'indirizzo 004015A3. Qual è il valore del registro EDX? (2) Eseguite 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**

ANALISI E VALUTAZIONE

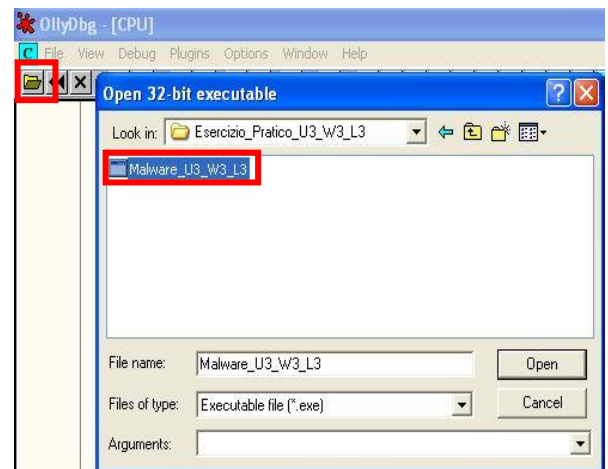
Come da task andiamo ad analizzare il malware che ha locazione nella cartella **Esercizio_Pratico_U3_W3_L3** sul Desktop. Il Malware ha nome **Malware_U3_W3_L3**:



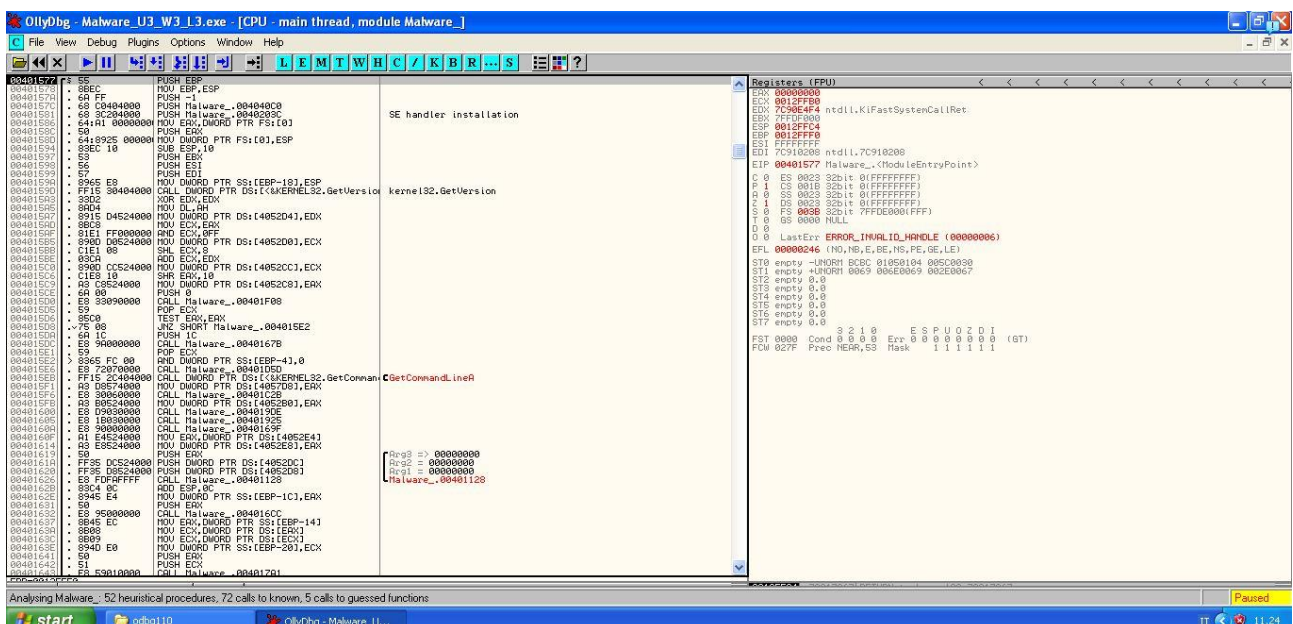
Successivamente andiamo ad aprire il tool che ci servirà per andare ad analizzare il Malware nella fase dinamica avanzata. Il tool è Ollydbg il quale ha funzionalità di analisi del malware mentre esso è in esecuzione, sfruttando i breakpoint per fermare temporaneamente l'esecuzione e recuperare informazioni sullo stato delle variabili, della memoria e dei registri:



All'apertura del tool, andiamo ad aprire il file a noi interessato direttamente dall'icona della cartella:



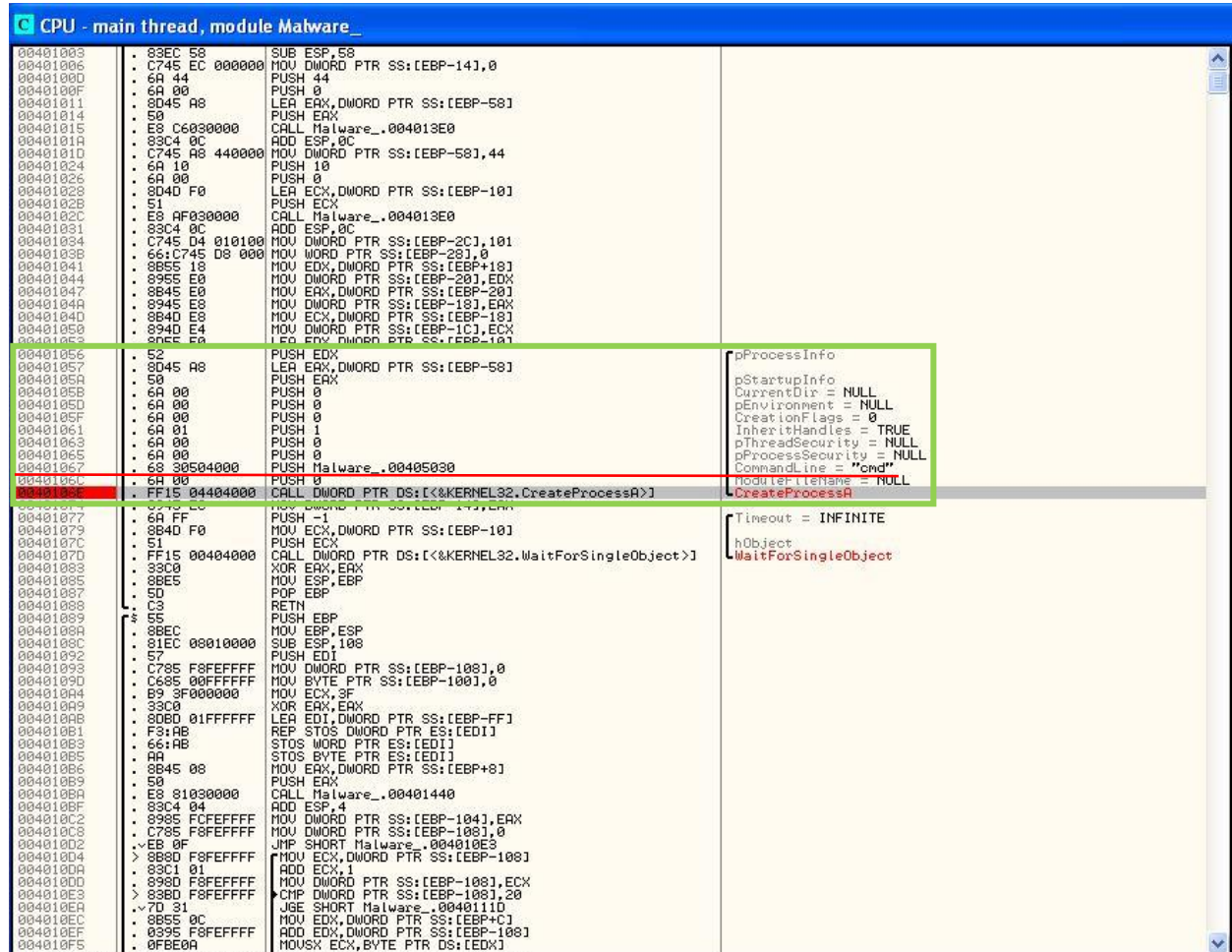
Dopo aver selezionato il file avremo una grafica di questo tipo:



Passiamo ora nel rispondere ai quesiti della task:


1. INDICARE IL VALORE DEL PARAMETRO ALLA LOCAZIONE 0040106E

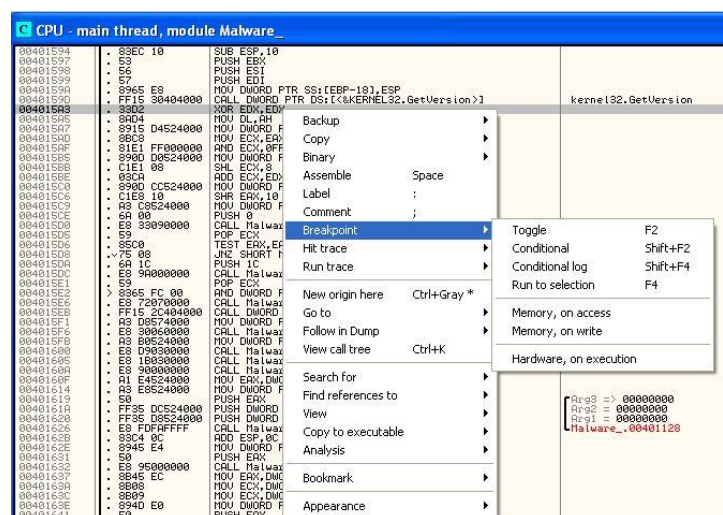
Come richiesto dalla task rechiamoci alla locazione di memoria indicata, la quale troveremo una funzione di chiamata <<**call ...**>>, il parametro della funzione a noi interessato è denominato <<**CommandLine**>>, come da immagine seguente possiamo affermare che il valore del parametro indicato è relativamente <<**cmd**>>:



2. INDICARE IL VALORE DEL REGISTRO EDX ALLA LOCAZIONE 004015A3

Come primo passaggio utilizziamo la funzionalità del breakpoint software impostandolo alla locazione indicata sopra, per farlo clicchiamo con il tasto destro e andiamo sulla sezione *breakpoint* → *toggle*.

Possiamo notare che il breakpoint è
Stato inserito in quanto la locazione
Verrà colorata in rosso: 



Il valore del registro **EDX** verrà visualizzato sulla destra, alla finestra *Registers (MMX)*:

```
Registers (MMX)
EAX 0A280105
ECX 7EED0000
EDX 00000A28
EBX 77F1434C
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 7FFDE000(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)
MM0 0105 0104 005C 0030
MM1 006E 0069 002E 0067
MM2 0000 0000 0000 0000
MM3 0000 0000 0000 0000
MM4 0000 0000 0000 0000
MM5 0000 0000 0000 0000
MM6 0000 0000 0000 0000
MM7 0000 0000 0000 0000
```

Il suo valore esadecimale è **0000A28** che in decimale diventa **2600**.

3. EFFETTUARE UNO STEP-INTO, INDICARE IL VALORE DEL REGISTRO EDX

Dopo aver ottenuto il valore del registro precedente andiamo ad eseguire uno step-into dall'icona indicata in figura, il quale permette di analizzare la locazione di memoria successiva a quella analizzata precedentemente, e andiamo a notare come il valore del registro è cambiato:

```
OllyDbg - Malware_U3_W3_I3.exe
File View Debug Plugins Options Window Help
[Icons] [L E M T W H C / K B R ... S] [Icons] [?]
CPU - main thread, module Malware_
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 . 64:01 00000000 MOV EBX,DWORD PTR FS:[0]
0040159C . 50 PUSH EAX
0040159D . 64:8925 00000000 MOV DWORD PTR FS:[0],ESP
00401594 . 8BEC 10 SUB ESP,10
00401597 . 59 PUSH EBX
00401598 . 58 PUSH ESI
00401599 . 57 PUSH EDI
0040159A . 8965 E8 MOV DWORD PTR SS:[EBP-10],ESP
0040159D . FF15 30404000 CALL DWORD PTR DS:[<&KERNEL32.GetVersion]
0040159E . 33D2 XOR EDX,EDX
0040159F . 80D4 MOV DL,AH
004015A7 . 8915 04524000 MOV DWORD PTR DS:[4052D4],EDX
004015AD . 8BC3 MOV ECX,EBX
004015AF . 01E1 FF000000 AND ECX,0FF
004015B5 . 8900 D0524000 MOV DWORD PTR DS:[4052D0],ECX
```

Il valore del registro EDX viene modificato in 0:

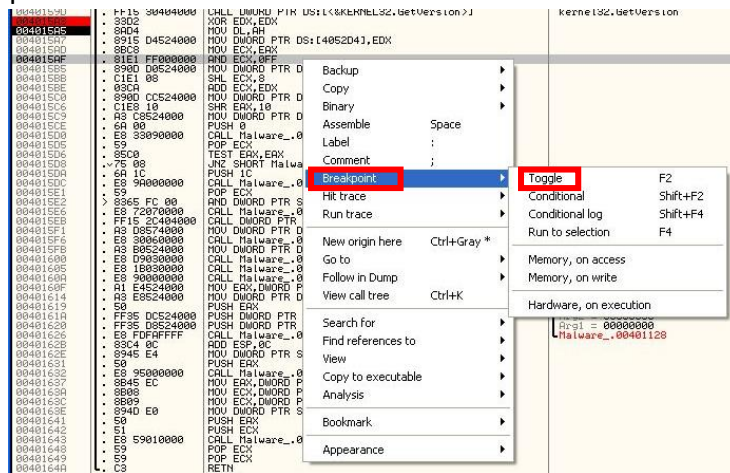
4. / 5. MOTIVA LA RISPOSTA E L'ISTRUZIONE ESEGUITA:

Il risultato ottenuto è dovuto al fatto che la locazione di memoria precedente contiene un'istruzione del tipo **XOR** la cui funzione di questo operatore logico è di mettere a confronto destinatario e sorgente, e se i due valori sono identici l'istruzione darà come risultato 0, come in questo caso.

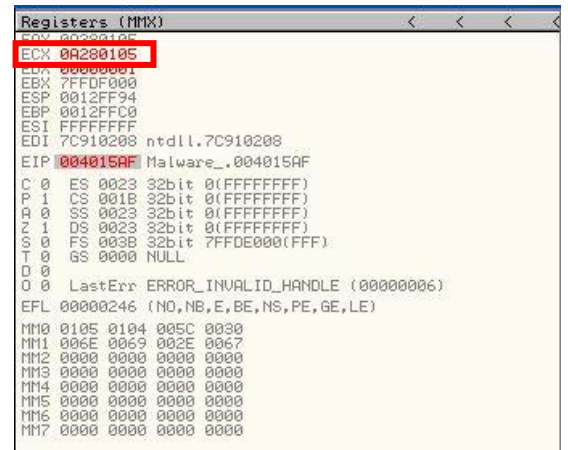
```
Registers (MMX)
EAX 0A280105
ECX 7EED0000
EDX 00000000
EBX 77F1434C
ESP 0012FF94
EBP 0012FFC0
ESI FFFFFFFF
EDI 7C910208 ntdll.7C910208
EIP 004015A5 Malware_.004015A5
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 1 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDE000(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)
MM0 0105 0104 005C 0030
MM1 006E 0069 002E 0067
MM2 0000 0000 0000 0000
MM3 0000 0000 0000 0000
MM4 0000 0000 0000 0000
MM5 0000 0000 0000 0000
MM6 0000 0000 0000 0000
MM7 0000 0000 0000 0000
```

6. INDICARE IL VALORE DEL REGISTRO ECX ALLA LOCAZIONE 004015AF

Come nel punto 2. Anche in questo caso andiamo ad utilizzare il breakpoint per andare a conoscere il valore del registro ECX alla locazione indicata. Il procedimento è lo stesso del punto 2.:

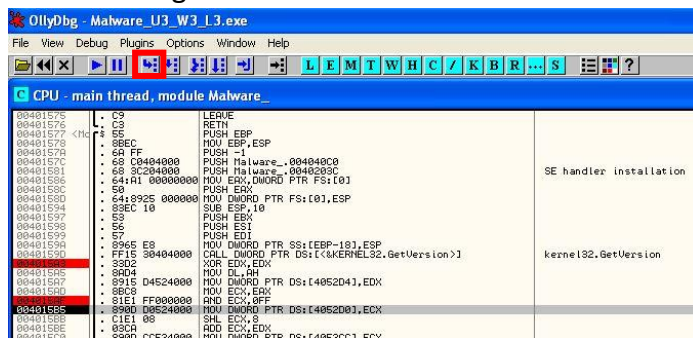


Il valore del registro ECX è in esadecimale **0A280105**, che in decimale diventa **170393861**:

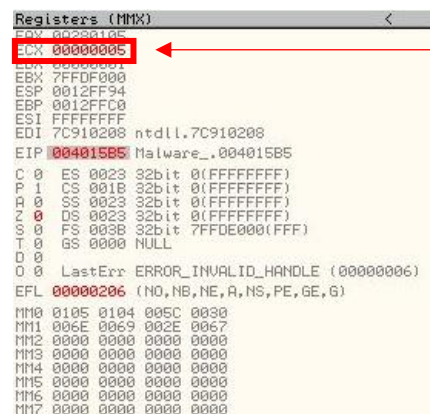


7. EFFETTUARE UNO STEP-INTO, INDICARE IL VALORE DEL REGISTRO ECX

Come il punto 3. Andiamo ad effettuare uno step-into e andiamo a vedere come cambia il valore del registro indicato:



Nell'istruzione successiva il valore del registro viene modificato in esadecimale **00000005**, che in decimale diventa **5**:



8. SPIEGARE QUALE ISTRUZIONE VIENE ESEGUITA

L'istruzione che viene eseguita in questo caso è denominata con **<<AND ..>>**. Questa istruzione è un prodotto logico che esegue la and logica dei 2 operandi (destinazione e sorgente). Il risultato è lasciato nell'operando di destinazione, al posto di quello di partenza.

9. BONUS: IPOTESI SUL FUNZIONAMENTO DEL MALWARE

Possiamo supporre che il Malware ha una funzione di Backdoor dovuto al fatto che nelle istruzioni dell'immagine seguente il malware va ad instaurare una connessione attraverso il Socket:

00401284	> 6A 00	PUSH 0	[Flags = 0 Group = 0 pWSAProtocol = NULL Protocol = IPPROTO_TCP Type = SOCK_STREAM Family = AF_INET WSASocketA
00401286	. 6A 00	PUSH 0	
00401288	. 6A 00	PUSH 0	
0040128A	. 6A 06	PUSH 6	
0040128C	. 6A 01	PUSH 1	
0040128E	. 6A 02	PUSH 2	
00401290	. FF15 A0404000	CALL DWORD PTR DS:[<&WS2_32.WSASocketA>]	
00401296	. 8985 FCFCFFF	MOV DWORD PTR SS:[EBP-304],EAX	