



# Esercizio 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.

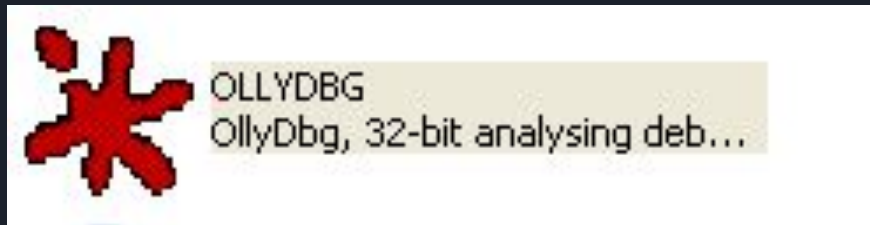
1. All'indirizzo 0040106E il Malware effettua una chiamata di funzione alla funzione «CreateProcess». Qual è il valore del parametro «CommandLine» che viene passato sullo stack?
2. Inserite un breakpoint software all'indirizzo 004015A3. Qual è il valore del registro EDX? Eseguite a questo punto uno «step-into». Indicate qual è ora il valore del registro EDX motivando la risposta. Che istruzione è stata eseguita?
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.

# Informazioni generali:

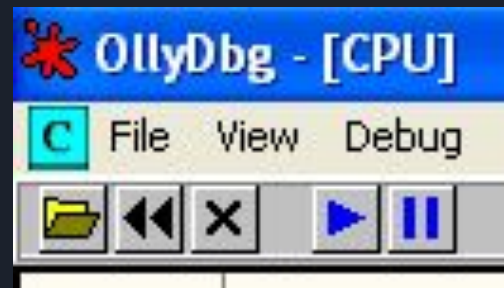
Per lo svolgimento dell'esercizio di oggi andremo ad utilizzare il programma OllyDBG presente nella macchina di malware analysis fornita.

OllyDBG è un debugger per sistemi Microsoft Windows, disponibile solo in versione a 32 bit. OllyDbg traccia i registri, riconosce le funzioni e i parametri passati alle principali librerie standard, le chiamate alle API del sistema operativo, eventuali salti condizionali, tabelle, costanti, variabili e stringhe.

Per avviare il programma apriamo la cartella "odbg110" presente nel desktop e facciamo doppio click sull'eseguibile all'interno. Comunque riportato in figura qua sotto.



Una volta avviato il programma dobbiamo caricare il file del malware da analizzare, per fare questo andiamo a cliccare sull'icona della cartella gialla presente in alto a sinistra del programma.



Adesso andiamo a selezionare il file Malware\_U3\_W3\_L3 contenuto nella cartella Esercizio\_Pratico\_U3\_W3\_L3 presente nel Desktop. Una volta selezionato il file premiamo sul tasto Open per caricarlo nel programma e per cominciare ad analizzarlo.



# Svolgimento del punto 1:

Adesso che abbiamo caricato il file malware all'interno del nostro debugger possiamo cominciare a navigare il codice alla ricerca dell'indirizzo di memoria 0040106E cosicché noi possiamo andare a rispondere alla prima domanda. Una volta individuato l'indirizzo di memoria possiamo vedere che il Malware effettua una chiamata di funzione alla funzione «CreateProcess». Ci viene chiesto adesso qual è il valore del parametro CommandLine che viene passato sullo stack per eseguire la funzione. Per rispondere ci basta osservare qualche riga sopra la chiamata di funzione, più precisamente all'indirizzo di memoria 00401067 dove tale parametro viene passato e possiamo vedere che presenta la dicitura CommandLine = "cmd", di conseguenza possiamo affermare che il valore del parametro CommandLine è appunto "cmd".

il comando cmd avvia una nuova istanza dell'interprete dei comandi, Cmd.exe. Se utilizzata senza parametri, cmd vengono visualizzate le informazioni sul copyright e versione del sistema operativo.

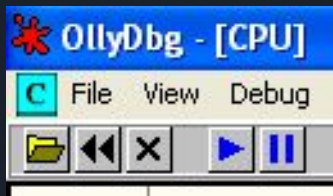
00401067	. 68 30504000	PUSH Malware_.00405030	ProcessName = "cmd"
0040106C	. 6A 00	PUSH 0	ModuleFileName = NULL
0040106E	. FF15 04404000	CALL DWORD PTR DS:[&KERNEL32.CreatePro	CreateProcessA
00401074	. 8945 EC	MOV DWORD PTR SS:[EBP-14],EAX	

## Svolgimento del punto 2:

Per rispondere al secondo quesito dobbiamo innanzitutto rintracciare l'indirizzo di memoria 004015A3, una volta individuato procediamo ad inserire un breakpoint a questo indirizzo nel seguente modo: Tasto destro sull'indirizzo di memoria > Breakpoint > Toggle

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

Una volta inserito il breakpoint facciamo partire il programma premendo il pulsante d'avvio a forma di triangolo situato nella barra degli strumenti in cima al programma.



## Svolgimento del punto 2:

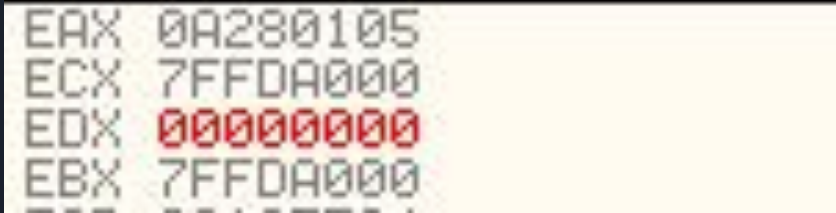
Dopo aver premuto il pulsante di avvio il malware si eseguirà e si interromperà all'istruzione immediatamente precedente rispetto al punto in cui noi abbiamo inserito il nostro breakpoint. Ci viene chiesto adesso di verificare il valore del registro EDX, il valore di questo registro è possibile vederlo nella finestra Registers (FPU) a destra del nostro codice, come possiamo osservare il registro EDX ha valore 00000A28 a questo punto dell'esecuzione.

EAX	0A280105
ECX	7FFDA000
EDX	00000A28
EBX	7FFDA000



Adesso eseguiamo uno Step-Into premendo il pulsante nell'immagine mostrata qua sopra per far eseguire le istruzioni all'indirizzo di memoria in cui noi abbiamo posizionato il nostro breakpoint. Dopo aver fatto ciò andiamo a verificare se il valore del registro EDX è cambiato.

## Svolgimento del punto 2:



```
EAX 0A280105
ECX 7FFDA000
EDX 00000000
EBX 7FFDA000
```

L'immagine qua sopra mostra il cambiamento del valore contenuto nel registro EDX dopo che l'istruzione `XOR EDX, EDX` è stata eseguita. Il valore di EDX corrisponde adesso a 00000000.

Il valore del registro EDX è stato azzerato dal peculiare comportamento dell'operatore logico XOR quando compara un elemento a se stesso, infatti siccome XOR restituisce 1 solamente quando i due valori comparati sono diversi fra loro, quando si compara un valore con se stesso si otterrà sempre l'azzeramento del valore comparato.



## Svolgimento del punto 3:

Esattamente come abbiamo fatto per il punto due andiamo ad inserire un nuovo breakpoint, questa volta all'indirizzo di memoria 004015AF e andiamo a verificare il valore del registro ECX prima che le istruzioni a questo indirizzo di memoria vengano eseguite.

004015AD	. 8BC8	MOV ECX,EAX
004015AF	. 81E1 FF000000	AND ECX,0FF
004015B5	. 890D 00524000	MOV DWORD PTR DS:[4052D0],ECX
004015BB	. C1F1 08	SHL ECX,8

Registers (FPU)	
EAX	0A280105
ECX	0A280105
EDX	00000001
EBX	77F14332

Come possiamo vedere il valore del registro ECX prima dell'esecuzione delle istruzioni è equivalente a 0A280105. Eseguiamo adesso uno Step-Into, ovvero andiamo ad eseguire le istruzioni all'indirizzo di memoria 004015AF e vediamo come cambia il valore di ECX.

## Svolgimento del punto 3:



Registers (FPU)	
EAX	0A280105
ECX	00000005
EDX	00000001
EBP	75550000

Come possiamo vedere, dopo aver eseguito le istruzioni AND ECX, 0FF il valore contenuto nel registro è stato cambiato a 00000005. Andiamo adesso a capire come siamo arrivati a questo valore. AND è un operatore logico binario che restituisce 1 solamente quando entrambi i bit comparati sono uguali a 1, quindi essenzialmente l'operazione che viene svolta è:

- tradurre in codice binario il valore esadecimale contenuto nel registro ECX ed il valore FF
- compararli con l'operatore logico binario AND
- Inserire il risultato dell'operazione all'interno del registro ECX ritradotto in esadecimale.