

Consegna S11-L2

Analisi con ida pro

Traccia:

Lo scopo dell'esercizio di oggi è di acquisire esperienza con IDA, un tool fondamentale per l'analisi statica. A tal proposito, con riferimento al malware chiamato «Malware_U3_W3_L2» presente all'interno della cartella «Esercizio_Pratico_U3_W3_L2» sul desktop della macchina virtuale dedicata all'analisi dei malware, rispondere ai seguenti quesiti, utilizzando IDA Pro.

1. Individuare l'indirizzo della funzione DLLMain
2. Dalla scheda «imports» individuare la funzione «gethostbyname». Qual è l'indirizzo dell'import?
3. Quante sono le variabili locali della funzione alla locazione di memoria 0×10001656?
4. Quanti sono, invece, i parametri della funzione sopra?



1

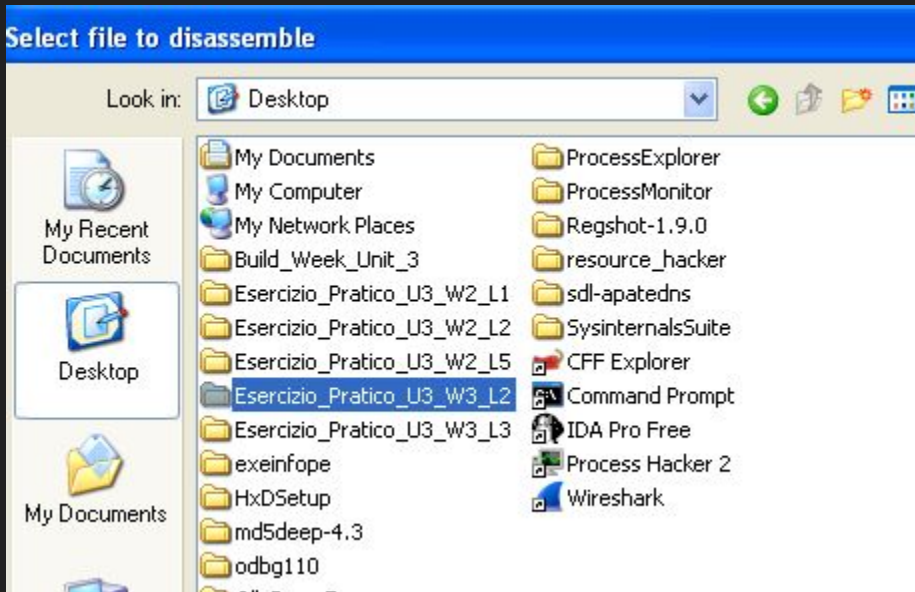


2



4

Come prima cosa, ovviamente, assicuriamoci, con gli step visti nelle precedenti lezioni, che la macchina virtuale sia sicura.



3

Fatto ciò iniziamo avviando il programma **IDA Pro Free** dal desktop con un semplice **doppio click**, fatto ciò clicchiamo sull'icona della cartella per selezionare, sempre da desktop, la cartella di nostro interesse, ovvero **Esercizio_Pratico_U3_W3_L2**, una volta aperta selezioniamo il formato **.dll** (**Malware_U3_W3_L2.dll**)

Per rispondere al primo quesito cerchiamo **DLLMain** nella sezione grafica del codice, una volta trovata la sezione a noi interessata premiamo sulla barra spaziatrice così da trovarci sull'interfaccia testuale del codice che ci aiuterà ad analizzarlo più facilmente, possiamo notare che l'indirizzo è **1000D02E**

```

; BOOL __stdcall DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPVOID lpvReserved)
_DllMain@12 proc near

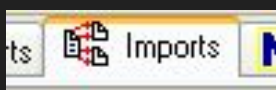
hinstDLL= dword ptr 4
fdwReason= dword ptr 8
lpvReserved= dword ptr 0Ch

mov     eax, [esp+fdwReason]
dec     eax
jnz     loc_1000D107

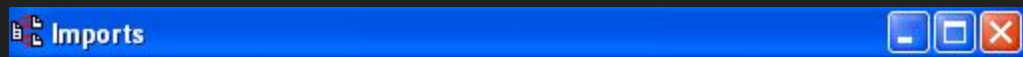
```

Barra spaziatrice

[illegible]



1



Address	Ordinal	Name	Library
100162E8		_strtime	MSVCRT
10016258		_strupr	MSVCRT
100162E0		_vsnprintf	MSVCRT
10016268		abs	MSVCRT
100162B4		atoi	MSVCRT
100163F4	3	closesocket	WS2_32
100163...	4	connect	WS2_32
100162A4		fclose	MSVCRT
10016274		fopen	MSVCRT
100162E4		fprintf	MSVCRT
10016234		fread	MSVCRT
100162...		free	MSVCRT
100162...		fseek	MSVCRT
10016278		ftell	MSVCRT
100162A0		fwrite	MSVCRT
100163...	52	gethostname	WS2_32
100163E4	9	htons	WS2_32
100163C8	11	inet_addr	WS2_32
100163...	12	inet_ntoa	WS2_32
1001624C		isdigit	MSVCRT

2

Per quanto riguarda il secondo quesito andremo a selezionare la voce imports in alto e poi scorriamo fino a trovare 'gethostname' troveremo così l'indirizzo dell'import, ovvero 100163CC

```

.text:10001656 ; !!!!!!!!!!!!!!! SUBROUTINE !!!!!!!!!!!!!!!
.text:10001656
.text:10001656
.text:10001656 ; DWORD __stdcall sub_10001656(LPVOID)
.text:10001656 sub_10001656    proc near                ; DATA XREF: DllMain(x,x,x)+C8↓o
.text:10001656
.text:10001656 var_675          = byte ptr -675h
.text:10001656 var_674          = dword ptr -674h
.text:10001656 hModule          = dword ptr -670h
.text:10001656 timeout          = timeval ptr -66Ch
.text:10001656 name            = sockaddr ptr -664h
.text:10001656 var_654          = word ptr -654h
.text:10001656 in              = in_addr ptr -650h
.text:10001656 Parameter        = byte ptr -644h
.text:10001656 CommandLine      = byte ptr -63Fh
.text:10001656 Data              = byte ptr -638h
.text:10001656 var_544          = dword ptr -544h
.text:10001656 var_50C          = dword ptr -50Ch
.text:10001656 var_500          = dword ptr -500h
.text:10001656 var_4FC          = dword ptr -4FCh
.text:10001656 readfds          = fd_set ptr -4BCh
.text:10001656 phkResult        = HKEY__ ptr -3B8h
.text:10001656 var_3B0          = dword ptr -3B0h
.text:10001656 var_1A4          = dword ptr -1A4h
.text:10001656 var_194          = dword ptr -194h
.text:10001656 WSADATA          = WSADATA ptr -190h
.text:10001656 arg_0            = dword ptr 4

```

Per rispondere agli ultimi due quesiti ricerchiamo il codice di memoria '10001656' ci troveremo davanti alla seguente pagina dove possiamo vedere che 20 funzioni sono con valore negativo rispetto all'offset di EBP mentre solo una e' positiva, chiamata **arg_0** da IDA Pro