

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?

```
.text:1000D02B
.text:1000D02E
.text:1000D02E ; !!!!!!!!!!!!!!! S U B R O U T I N E !!!!!!!!!!!!!!!
.text:1000D02E
.text:1000D02E ; BOOL __stdcall DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPOVOID lpvReserved)
.text:1000D02E _DllMain@12      proc near                ; CODE XREF: DllEntryPoint+4B↓p
.text:1000D02E                                           ; DATA XREF: sub_100110FF+2D↓o
.text:1000D02E
.text:1000D02E hinstDLL      = dword ptr  4
.text:1000D02E fdwReason     = dword ptr  8
.text:1000D02E lpvReserved   = dword ptr 0Ch
.text:1000D02E
* .text:1000D02E      mov     eax, [esp+fdwReason]
* .text:1000D032      dec     eax
* .text:1000D033      jnz     loc_1000D107
```

L'indirizzo della funzione main è: 1000D02E

Imports			
Address	Ordinal	Name	Library
10016398		OpenInputDesktop	USER32
1001639C		GetUserObjectInformationA	USER32
100163A4		waveInReset	WINMM
100163A8		waveInOpen	WINMM
100163...		waveInClose	WINMM
100163B0		waveInUnprepareHeader	WINMM
100163B4		waveInPrepareHeader	WINMM
100163B8		waveInAddBuffer	WINMM
100163...		waveInStart	WINMM
100163C4	18	select	WS2_32
100163C8	11	inet_addr	WS2_32
100163...	52	gethostbyname	WS2_32
100163...	12	inet_ntoa	WS2_32
100163...	16	recv	WS2_32
100163...	19	send	WS2_32
100163...	4	connect	WS2_32
100163E0	15	ntohs	WS2_32
100163E4	9	htons	WS2_32
100163E8	21	setsockopt	WS2_32
100163...	116	WSACleanup	WS2_32
100163F0	115	WSAStartup	WS2_32
100163F4	3	closesocket	WS2_32
100163F8	23	socket	WS2_32
100163FC	111	WSAGetLastError	WS2_32
10016404		GetAdaptersInfo	iphlpapi
1001640C		CoTaskMemFree	ole32
10016410		CoInitialize	ole32
10016414		CoInitializeEx	ole32
10016418		CoCreateInstance	ole32
1001641C		CoUninitialize	ole32

L'indirizzo di "gethostbyname" è: 100163CC

```
IDA View-A
.text:10001656 ; SUBROUTINE
.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
.text:10001656
* .text:10001656 sub esp, 678h
* .text:1000165C push ebx
* .text:1000165D push ebp
* .text:1000165E push esi
* .text:1000165F push edi
```

A questo indirizzo (10001656) ci sono 20 variabili

C'è un solo argomento passato alla funzione, ha offset positivo rispetto a EBP, chiamato da IDA "arg_0"