

S11L2

Rosario Z.

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.

- Individuare l'indirizzo della funzione DLLMain.
- Dalla scheda «imports» individuare la funzione «gethostbyname». Qual è l'indirizzo dell'import?
- Quante sono le variabili locali della funzione alla locazione di memoria 0x10001656?
- Quanti sono, invece, i parametri della funzione sopra?

Traccia 1

Individuare l'indirizzo della funzione DLLMain

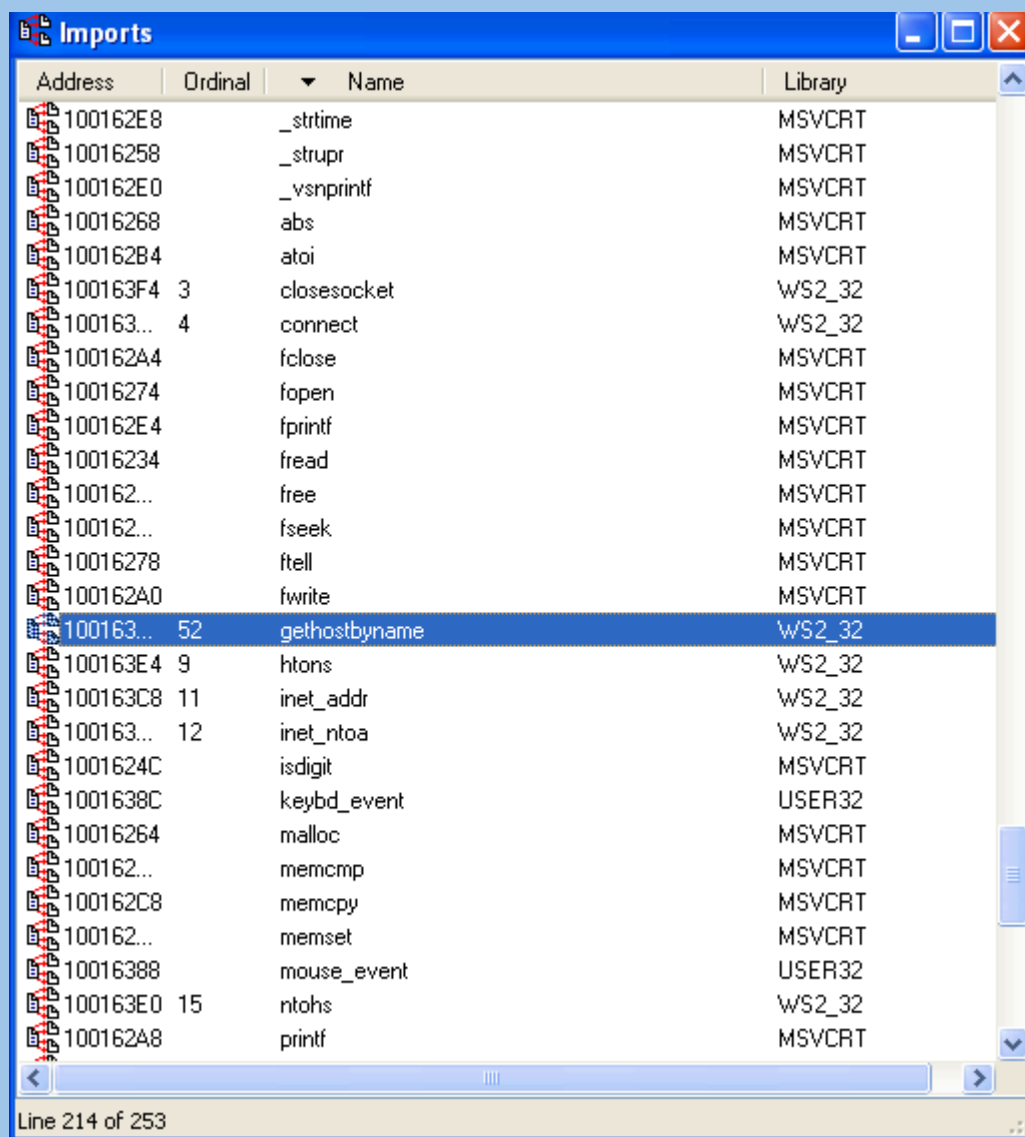
Al fine di trovare l'indirizzo della funzione DllMain, carichiamo l'eseguibile in IDA Pro. Una volta fatto, premiamo la barra per passare nella modalità testuale e recuperare l'indirizzo della funzione main che sarà: 1000D02E

```
.text:1000D02E
.text:1000D02E
.text:1000D02E ; BOOL __stdcall DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPOVOID lpvReserved)
.text:1000D02E _DllMain@12      proc near                                ; CODE XREF: DllEntryPoint+4B.jp
.text:1000D02E                                           ; DATA XREF: sub_100110FF+2D.jp
.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
.text:1000D039      mov     eax, [esp+hinstDLL]
.text:1000D03D      push    ebx
.text:1000D03E      mov     ds:hModule, eax
.text:1000D043      mov     eax, off_10019044
.text:1000D048      push    esi
.text:1000D049      add     eax, 0Dh
```

Traccia 2

Dalla scheda «imports» individuare la funzione «gethostbyname». Qual è l'indirizzo dell'import?

Apriamo la finestra degli «imports» da IDA Pro, e localizziamo la funzione cercata. «gesthostbyname» è all'indirizzo 100163CC, come mostrato in figura



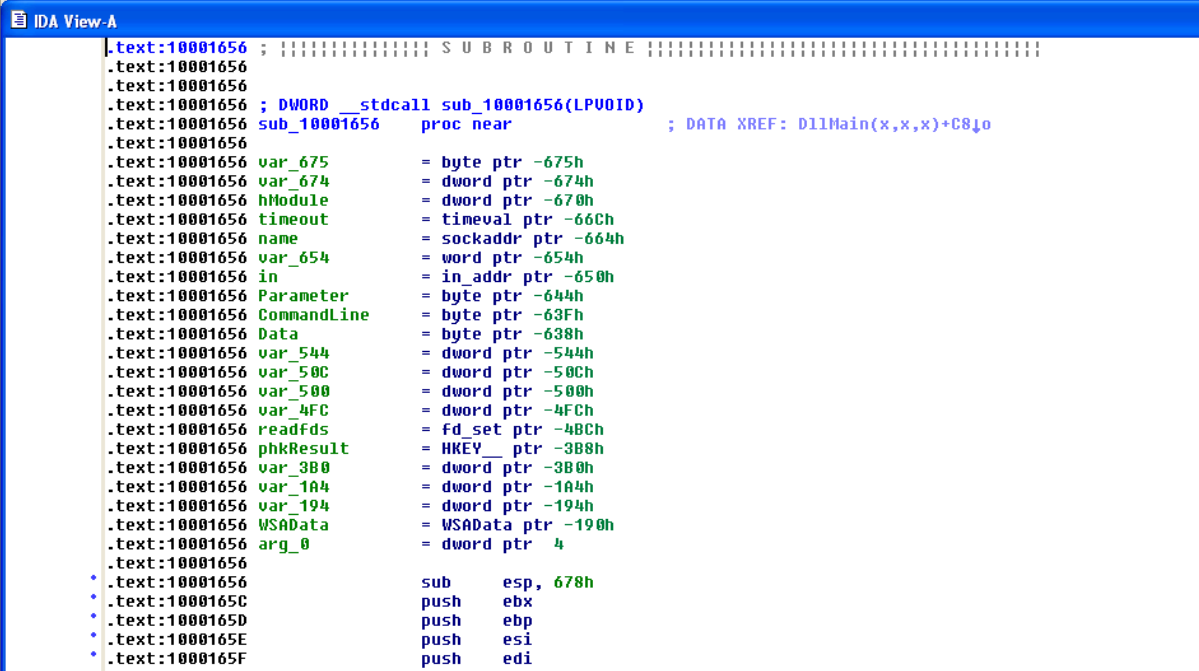
Traccia 3

- Quante sono le variabili locali della funzione alla locazione di memoria 0x10001656?

Quante sono le variabili locali della funzione alla locazione di memoria 10001656?

Per prima cosa bisogna spostarsi all'indirizzo ricercato tramite la ricerca o la barra laterale.

A questo indirizzo troviamo 20 variabili con offset negativo rispetto ad EBP



```
.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)+C8Jo
.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 -48Ch
.text:10001656 phkResult = HKEY__ ptr -388h
.text:10001656 var_380 = dword ptr -380h
.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:10001656 push ebx
* .text:10001656 push ebp
* .text:10001656 push esi
* .text:10001656 push edi
```

Traccia 4

Quanti parametri sono presenti nella funzione di cui sopra?

Dalla stessa figura, possiamo notare un solo argomento passato alla funzione, avente offset positivo rispetto ad EBP. IDA ha chiamato questo parametro «arg_0».