

# Esercizio S11 L2

## 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.

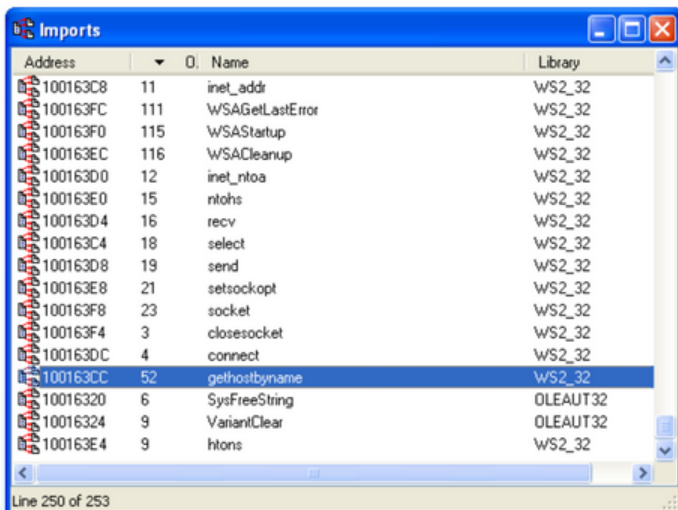
```

.text:100002E
.text:100002E ; :::::::::::::::::::::: SUBROUTINE ::::::::::::::::::::::::::::::
.text:100002E
.text:100002E
.text:100002E ; BOOL __stdcall DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPOVOID lpvReserved)
.text:100002E _DllMain@12 proc near ; CODE XREF: DllEntryPoint+4B↓p
.text:100002E ; DATA XREF: sub_100110FF+2D↓o
.text:100002E
.text:100002E hinstDLL = dword ptr 4
.text:100002E fdwReason = dword ptr 8
.text:100002E lpvReserved = dword ptr 0Ch
.text:100002E
* .text:100002E mov eax, [esp+fdwReason]

```

L'indirizzo della funzione `DLLMain` è **1000D02E**.

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



L'indirizzo dell'import è **100163CC**.

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

```
.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)+C840
.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 -4BCCh
.text:10001656 phkResult = HKEY__ ptr -3B8h
.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 sub esp, 678h
```

variabili locali

parametri

Possiamo notare **venti variabili** con offset negativo.

Quanti sono, invece, i parametri della funzione sopra?

Sempre dall'immagine sopra possiamo notare **solo un parametro** con offset positivo,

Inserire considerazioni macro livello sul malware.

Esplorando il codice assembly è facile notare che il malware esegue diverse operazioni, esegue e modifica file e cartelle, verifica versioni di hardware e software e molto altro. Inoltre si collega ad un server esterno. Si tratta di una backdoor con funzioni di C&C. La cosa è confermata da alcune righe come questa:

```
.text:1000438A lea eax, [ebp+buf]
.text:10004390 push offset aBackdoorServer ; "\r\n\r\n*****\r\n[Ba"...
.text:10004395 push eax ; char *
.text:10004396 call esi ; sprintf
.text:10004398 mov ebx, [ebp+s]
.text:1000439B lea eax, [ebp+buf]
.text:100043A1 push eax ; buf
```