S11/L2

Malware analysis: Analisi statica avanzata

Lo scopo dell'esercizio consiste nell'analizzare il malware chiamato **Malware_U3_W3_L2**, tramite il tool IDA Pro e rispondere ai seguenti quesiti:

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 0x10001656?
4. Quanti sono, invece, i parametri della funzione sopra?
5. Inserire altre considerazioni macro livello sul malware (comportamento)

Esercizio

1.
Una volta caricato il file su IDA Pro ci rechiamo nella finestra Functions e andiamo a cercare la funzione **DLLMain**.
In questa finestra ci verranno fornite tutte le informazioni della funzione, compreso il suo indirizzo che nel nostro caso è **1000D02E**

```
.text:1000D02E
.text:1000D02E ; ||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||||||||||||||
.text:1000D02E
.text:1000D02E
.text:1000D02E ; BOOL __stdcall DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPVOID 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]
```

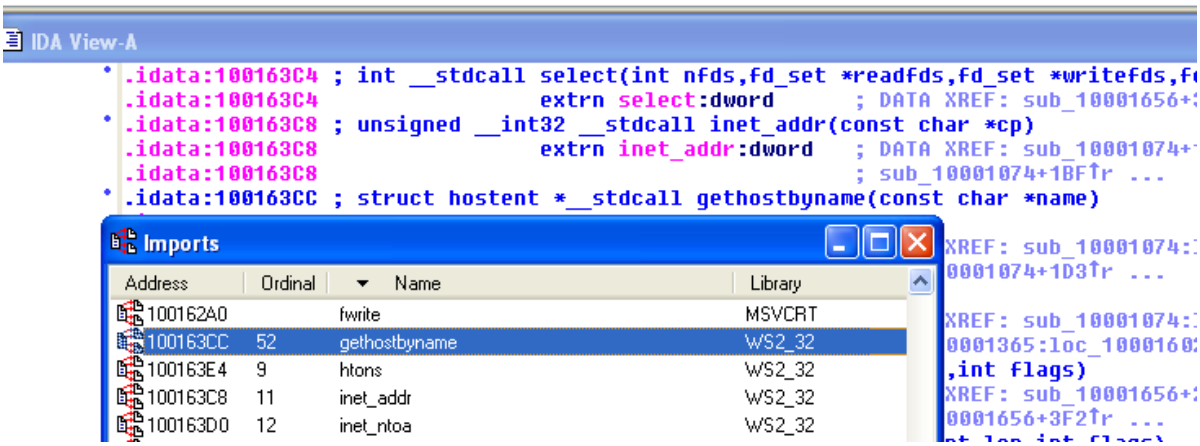
Functions window

Function name	Segment	Start	Length	R	F	L	S	B	T
DllMain(x,x,x)	.text	1000D02E	000000DF	R	.	.	.	.	T
sub_10011DB0	.text	10011DB0	000000E2	R	.	.	.	B	.
sub_1000B26C	.text	1000B26C	000000E7	R	.	.	.	B	.

2.

Per rispondere a questo quesito ci basta recarci semplicemente nella finestra di Imports e cercare la funzione **gethostbyname.**

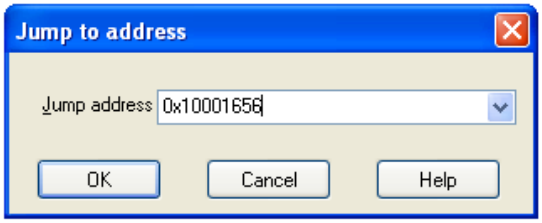
In questa finestra sono presenti le informazioni importate nell'eseguibile come libreria, nome e indirizzo di memoria, che in questo caso è **100163CC**.



3.4.

Per rispondere a questi due quesiti dobbiamo andare a cercare l'indirizzo di memoria 0x10001656 con la funzione "jump to address"

```
.text:10001656
.text:10001656 ; ||||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||||||||||||||
.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
.text:10001656
.text:10001656                 sub     esp, 678h
```



In questo caso abbiamo 20 variabili, le possiamo identificare perché hanno un offset negativo.
Mentre per quanto riguarda gli argomenti ne è presente solo uno, in questo caso è **arg_0** con un offset positivo.

5.
Il malware risulta essere una backdoor

```
xdoors_d:10093D50                 db '(1) Enter Current Directory ',27h,'%s',27h,0
xdoors_d:10093D73                 align 4
xdoors_d:10093D74 ; char aBackdoorServer[]
xdoors_d:10093D74 aBackdoorServer db 0Dh,0Ah              ; DATA XREF: sub_100042DB+B5↑o
xdoors_d:10093D74                 db 0Dh,0Ah
xdoors_d:10093D74                 db '*******************************',0Dh,0Ah
xdoors_d:10093D74                 db '[BackDoor Server Update Setup]',0Dh,0Ah
xdoors_d:10093D74                 db '*******************************',0Dh,0Ah
xdoors_d:10093D74                 db 0Dh,0Ah,0
xdoors_d:10093DDB                 align 4
xdoors_d:10093DDC ; char aWarn[]
```