

Esercitazione WEEK 23 D4

Analisi statica avanzata con IDA

Ettore Farris

Descrizione sintetica

Traccia:

Lo scopo dell'esercizio 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** (così com'è, in esadecimale).

Function name	Segment	Start	Length
BlockInput	.text	100111E2	00000006
CreateToolhelp32Snapshot	.text	100111C4	00000006
DllEntryPoint	.text	1001516D	0000009D
DllMain(x.x.x)	.text	1000D02E	000000DF
EnumProcessModules	.text	100111AC	00000006
GetAdaptersInfo	.text	100111B2	00000006
GetModuleFileNameExA	.text	100111A6	00000006
HandlerProc	.text	1000C9DF	00000077
ICClose	.text	100113D6	00000006

```
.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]
```

La funzione DLLMain si trova all'indirizzo 1000D02E. L'informazione la si ottiene dalla Functions Window.

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

Address	Ordinal	Name	Library
10016278		ftell	MSVCRT
100162A0		fwrite	MSVCRT
100163CC	52	gethostbyname	WS2_32
100163E4	9	htons	WS2_32
100163C8	11	inet_addr	WS2_32
100163D0	12	inet_ntoa	WS2_32

Dalla scheda *imports* vediamo che la funzione è all'indirizzo 100163CC.

- 3. Quante sono le variabili locali della **funzione** alla locazione di memoria 0x10001656?
- 4. Quanti sono, invece, i parametri della funzione sopra?

```
.text:10001656 ; ===== S U B R O U T I N E =====
.text:10001656
.text:10001656
.text:10001656 ; DWRD __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 hLibModule = dword ptr -670h
.text:10001656 timeout = timeval ptr -66Ch
.text:10001656 name = sockaddr ptr -664h
.text:10001656 var_654 = word ptr -654h
.text:10001656 Dst = dword ptr -650h
.text:10001656 Parameter = byte ptr -644h
.text:10001656 var_640 = byte ptr -640h
.text:10001656 CommandLine = byte ptr -63Fh
.text:10001656 Source = byte ptr -63Dh
.text:10001656 Data = byte ptr -638h
.text:10001656 var_637 = byte ptr -637h
.text:10001656 var_544 = dword ptr -544h
.text:10001656 var_50C = dword ptr -50Ch

.text:10001656 var_500 = dword ptr -500h
.text:10001656 Buf2 = byte ptr -4FCh
.text:10001656 readfds = fd_set ptr -48Ch
.text:10001656 phkResult = byte 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 WSADData = WSADData ptr -190h
.text:10001656 arg_0 = dword ptr 4
```

La funzione ha 23 variabili e 1 parametro chiamato *arg_0*. Questo lo si capisce dal segno dell'*offset* (negativo per le variabili e positivo per i parametri).

- 5. *Inserire altre considerazioni macro livello sul malware*

L'eseguibile è un trojan con una backdoor. Consultando le funzioni a cui accede, il malware ha accesso al file system e può manipolare i file, può manipolare il registro di sistema e può effettuare connessioni di rete. Inoltre intercetta i movimenti del mouse.