



Unità 3 Settimana 3

Compito 2

Individuare l'indirizzo della funzione DLLMain

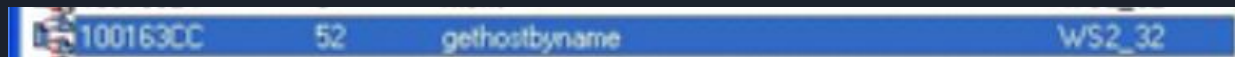
Per individuare l'indirizzo della funzione DLLMain, carichiamo il file eseguibile in IDA Pro. Dopo averlo fatto, passiamo alla modalità testuale e recuperiamo l'indirizzo della funzione main, che risulterà essere: 1000D02E.

```
.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
```

Indirizzo dell'import <<gethostbyname>>

Apriamo la finestra degli "imports" in IDA Pro e individuiamo la funzione cercata.

"Gethostbyname" si trova all'indirizzo 100163CC, come illustrato nella figura allegata.



Variabili locali della funzione alla locazione di memoria 10001656

Iniziamo localizzando l'indirizzo desiderato utilizzando la funzione di ricerca o la barra laterale. Una volta raggiunto l'indirizzo, notiamo la presenza di 20 variabili con offset negativo rispetto a EBP.

.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

Parametri presenti nella funzione

Nella stessa figura, è evidente che solo un argomento è passato alla funzione, con un offset positivo rispetto ad EBP. IDA lo ha identificato come "arg_0".

```
.text:10001656 ; SUBROUTINE
.text:10001656
.text:10001656 ; DWORD __stdcall sub_10001656(LPVOID)
.text:10001656 sub_10001656 proc near ; DATA XREF:
.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 -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 = WSADData ptr -190h
.text:10001656 arg_0 = dword ptr 4
.text:10001656
```



Considerazioni finali

- Presenza di funzioni di rete:** La presenza di una funzione come "Gethostbyname" suggerisce che il malware potrebbe coinvolgere attività di rete, come il contatto con server remoti per scopi di comando e controllo o per lo scambio di dati.
- Uso di argomenti nella funzione:** Il fatto che ci sia un argomento passato alla funzione può indicare che il malware riceve dati esterni o istruzioni tramite questo argomento, il che potrebbe essere correlato alle sue operazioni o al suo comportamento.
- Struttura di memoria:** La menzione di variabili con offset negativo rispetto a EBP suggerisce che il malware potrebbe utilizzare uno stack frame standard per la gestione delle variabili locali e degli argomenti della funzione.
- Analisi più approfondita:** Per fare considerazioni più precise e informate sul malware, sarebbe necessario un'analisi più approfondita delle sue funzionalità, del suo comportamento, delle sue tecniche di evasione e dei suoi obiettivi. Questo potrebbe richiedere l'ispezione del codice assembly, il reverse engineering della sua logica e l'analisi del traffico di rete generato dal malware.