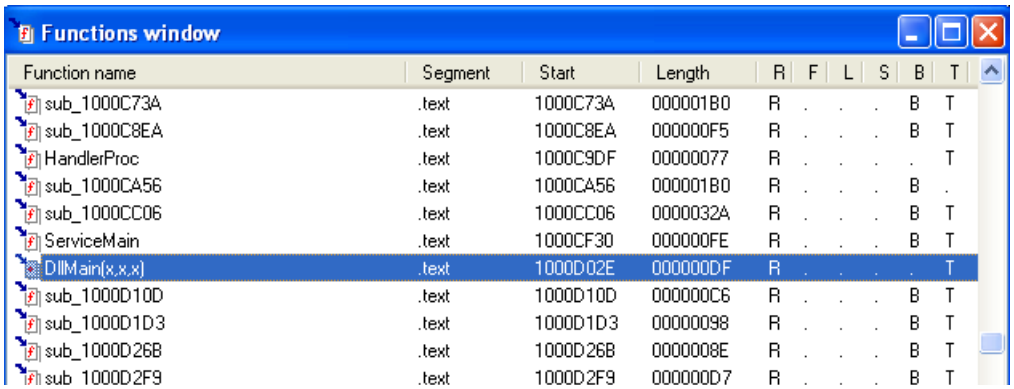
ANALISI STATICA AVANZATA CON IDA

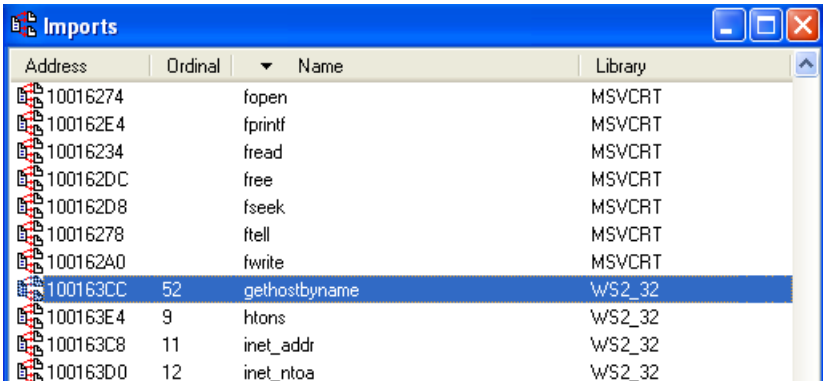
- Individuare l'indirizzo della funzione DLLMain.
- Dalla scheda “imports” individuare la funzione “gethostbyname”. Qual è l'indirizzo dell'import? Cosa fa la funzione?
- Quante sono le variabili locali della funzione alla locazione di memoria 0x10001656?
- Quanti sono, invece, i parametri della funzione sopra?
- Inserire altre considerazioni di macrolivello sul malware (comportamento).

Nell'esercizio di oggi andiamo a studiare il codice assembly di un malware tramite il tool IDA, eseguendo quindi un'analisi statica avanzata del programma.

IDA è un tool molto intuitivo, ci permette di individuare facilmente, tramite un'interfaccia grafica, le varie parti, funzioni, salti di un malware.
Usando la finestra "Function" possiamo individuare la funzione main del programma, chiamata DLLMain, alla locazione di memoria 1000D02E.



Un altro modo per trovare una funzione è attraverso la scheda "Import", che elenca tutte le funzioni importate dal malware.
La scheda ci da informazioni sul nome, la locazione di memoria e la liberia a cui appartiene. Nel nosto caso, la funzione "gethostbyname" si trova all'indirizzo 100163CC, nella libreria WS2_32.



La libreria WS2_32 viene usata per le comunicazioni di rete, per esempio creando socket e permettendo l'invio(ricezione di dati tramite il socket. La funzione gethostbyname permette di ricavare l'indirizzo IP dell'host, la macchina attaccata dal malware.

La scheda IDA View-A ci fornisce due tipi di visualizzazione, una in formato diagramma di flusso (che evidenzia le varie parti di codici e come esse sono collegate dai salti, sia condizionali che incondizionali), una in formato di codice, che da maggiori informazioni sulla sezione e sull'indirizzo di memoria.
Quando una nuova funzione viene chiamata, vengono elencati variabili e parametri locali della funzione, come nella figura sotto.

```
.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
```

La funzione sub_10001656 presenta 20 variabili, che IDA rende facilmente identificabili grazie all'offset negativo scritto nel codice.
Il parametro invece è uno solo, arg_0 (argument in inglese significa parametro).

In generale questo malware sembra poter fare molte cose interfacciandosi con il sistema operativo (inizializzare processi, creare, copiare o eliminare file, accedere alle chiavi di registro, creare un socket per la connessione di rete). Potrebbe quindi trattarsi di una backdoor che crea una shell sull'host target per poi accedervi da remoto e operare a piacimento.