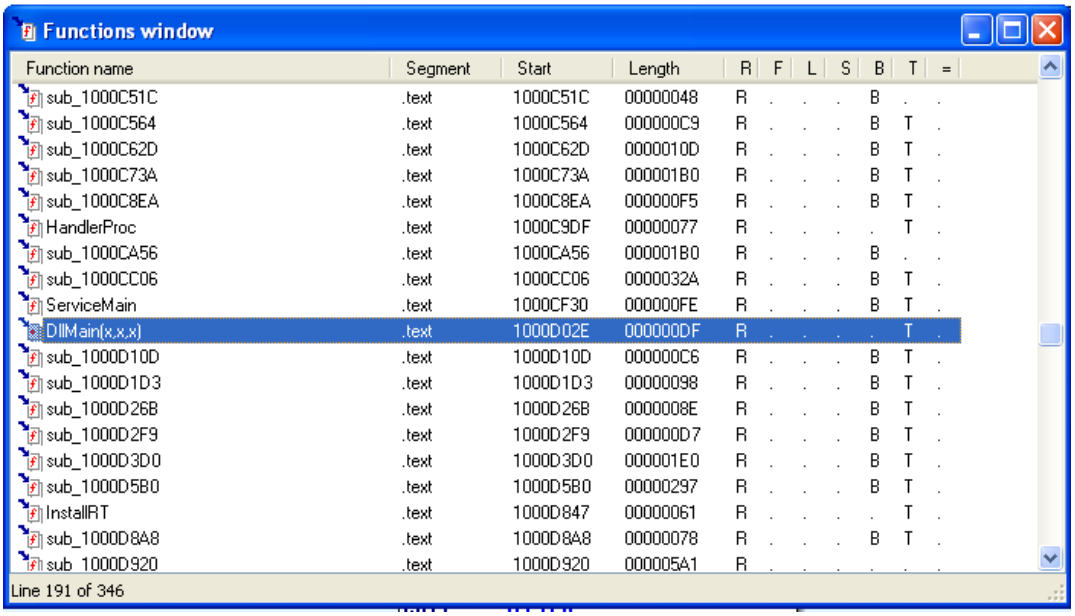
Analisi Statica Avanzata con IDA

Nell'esercizio di oggi useremo il **disassembler IDA** per effettuare un'**analisi statica avanzata** di un malware e, a scopo didattico, risponderemo ai seguenti quesiti:

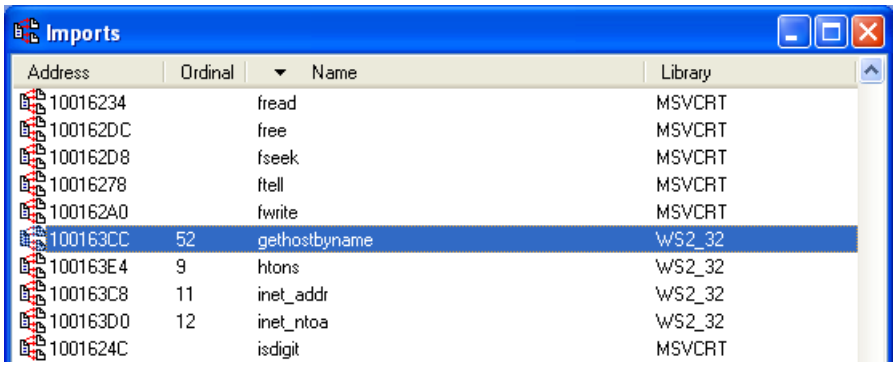
1. Individuare l'indirizzo della funzione **DLLMain** in esadecimale;
2. Dalla scheda «imports» individuare la funzione **gethostbyname**, il suo indirizzo e la funzione;
3. Quante sono le **variabili** locali della funzione alla locazione di memoria **0x10001656**;
4. Quanti sono i **parametri** della funzione sopra;
5. Inserire altre **considerazioni** macro livello sul **comportamento** del malware.

Eseguiamo ora l'analisi:

1. Come evidenziato nello screenshot qui a lato, la funzione **DLLMain** si trova all'indirizzo di memoria **1000D02E**.



2. Nell'immagine a destra vediamo la funzione **gethostbyname**, all'indirizzo di memoria **100163CC**. Il suo scopo è quello di **risolvere l'host** fornito cercando di ricavarne l'**indirizzo IP**; deduciamo quindi che il malware esaminato ha **funzioni di network**.



3. All'indirizzo di memoria **0x10001656**, come visibile nello screenshot a destra, identifichiamo ben **venti variabili** e **un parametro**. Le prime le riconosciamo in quanto hanno un **offset negativo** rispetto **EBP**.
4. Facendo riferimento alla stessa immagine, il **parametro** è invece identificabile in quanto ha **offset positivo** rispetto **EBP**; nello screenshot è l'ultimo elemento della lista.

```
var_675= byte ptr -675h
var_674= dword ptr -674h
hModule= dword ptr -670h
timeout= timeval ptr -66Ch
name= sockaddr ptr -664h
var_654= word ptr -654h
in= in_addr ptr -650h
Parameter= byte ptr -644h
CommandLine= byte ptr -63Fh
Data= byte ptr -638h
var_544= dword ptr -544h
var_50C= dword ptr -50Ch
var_500= dword ptr -500h
var_4FC= dword ptr -4FCh
readfds= fd_set ptr -4BCh
phkResult= HKEY__ ptr -3B8h
var_3B0= dword ptr -3B0h
var_1A4= dword ptr -1A4h
var_194= dword ptr -194h
WSAData= WSAData ptr -190h
arg_0= dword ptr  4
```

5. Considerando il codice esaminato, possiamo supporre che il malware in questione sia una **backdoor**: ne abbiamo riscontro anche nel codice stesso, come mostrato nella seguente immagine.

```
push    offset aBackdoorServer ; "\r\n\r\n*******************************\r\n[Ba"...
```

Ne abbiamo ulteriore conferma ottenendo l'**hash** del malware e confrontandolo su VirusTotal.