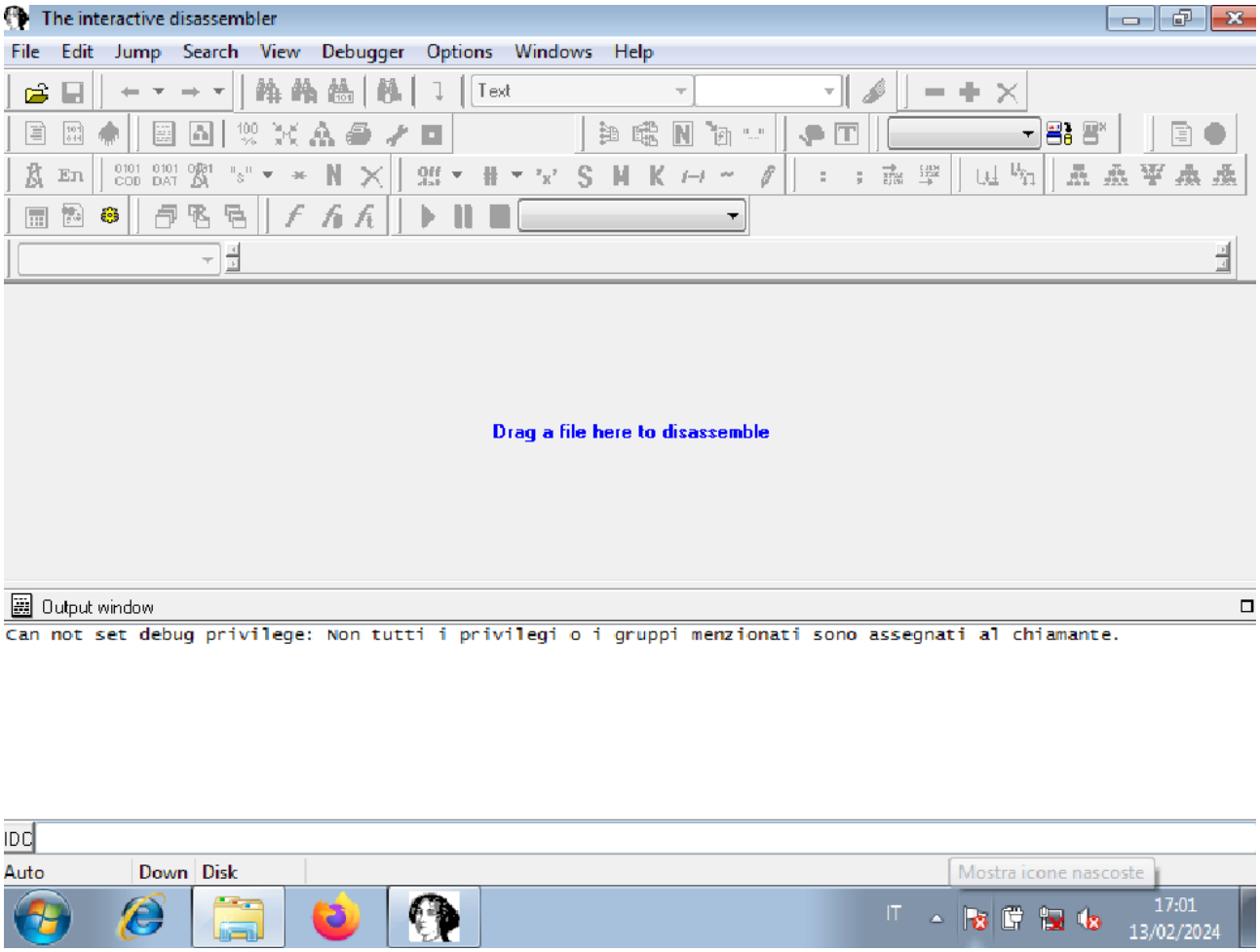
ESERCITAZIONE S11 L2

Analisi statica avanzata con IDA

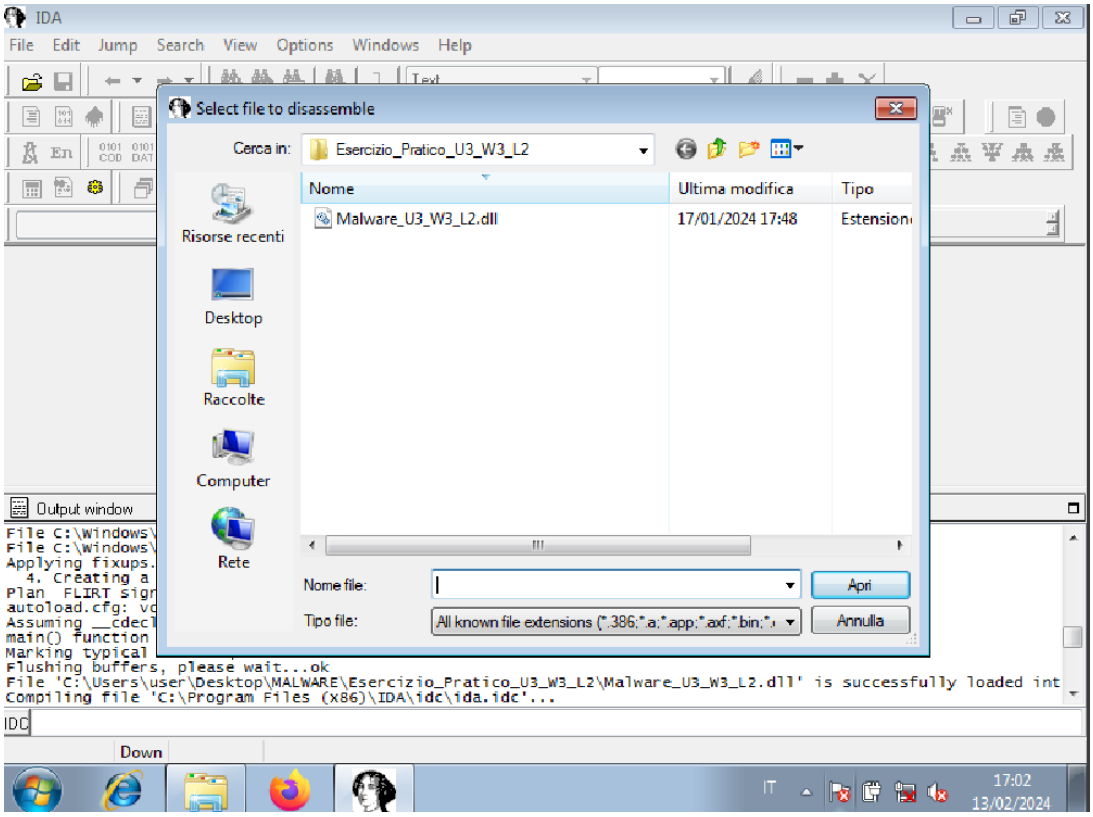
1. INDIVIDUARE L'INDIRIZZO DELLA FUNZIONE DLL MAIN
2. INDIVIDUARE LA FUNZIONE "GETHOSTBYNAME"
3. VARIBIALI LOCALI ALLA LOCAZIONE DI MEMORIA 0x10001656
4. PARAMETRI ALLA LOCAZIONE DI MEMORIA 0x10001656

Per effettuare un'analisi statica avanzata con *IDA* bisogna avviarla e selezionare il file da analizzare, seguendo la procedura sottoindicata.

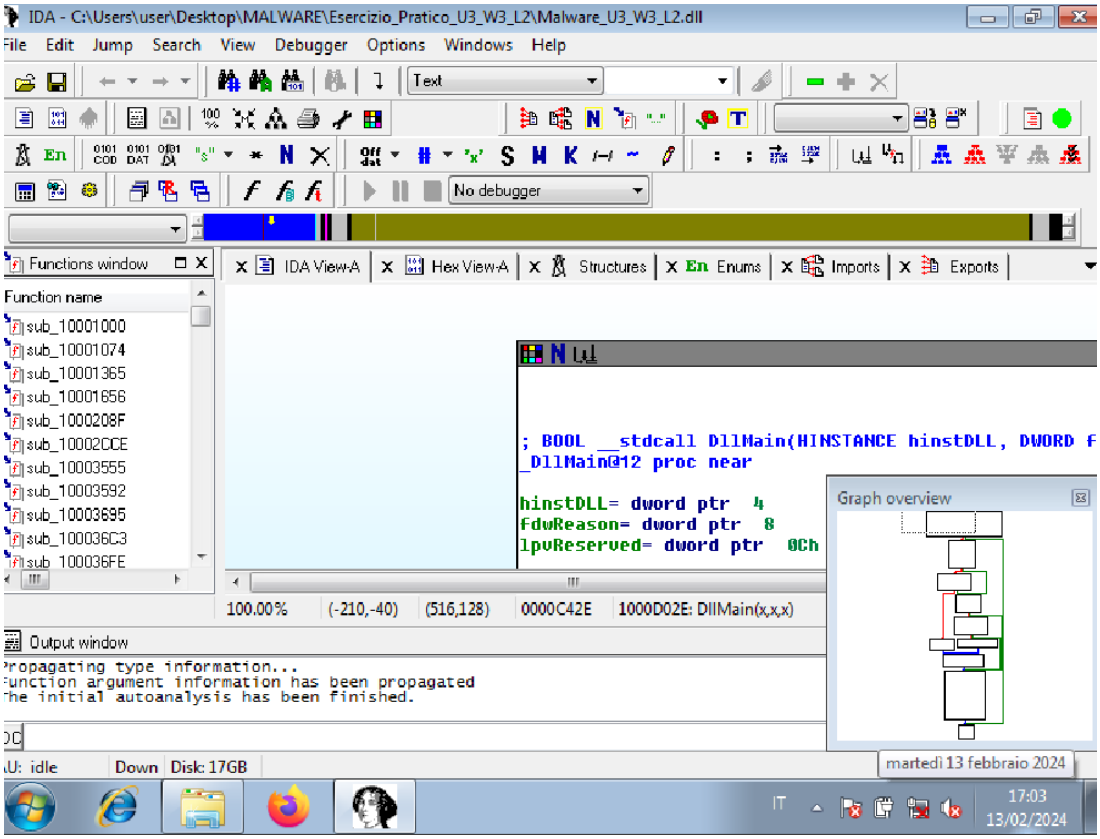
Avviamo IDA



Scegliamo il malware da analizzare, in questo caso MALWARE_U3_W3_L2.DLL

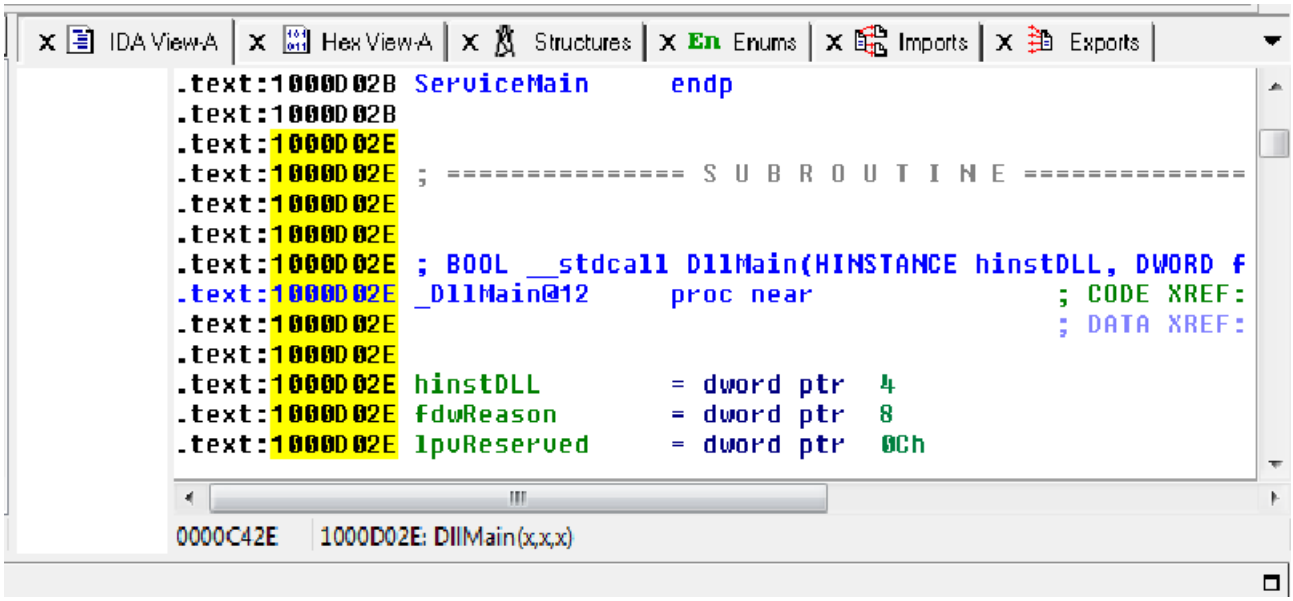


IDA pronta per effettuare l'analisi avanzata.



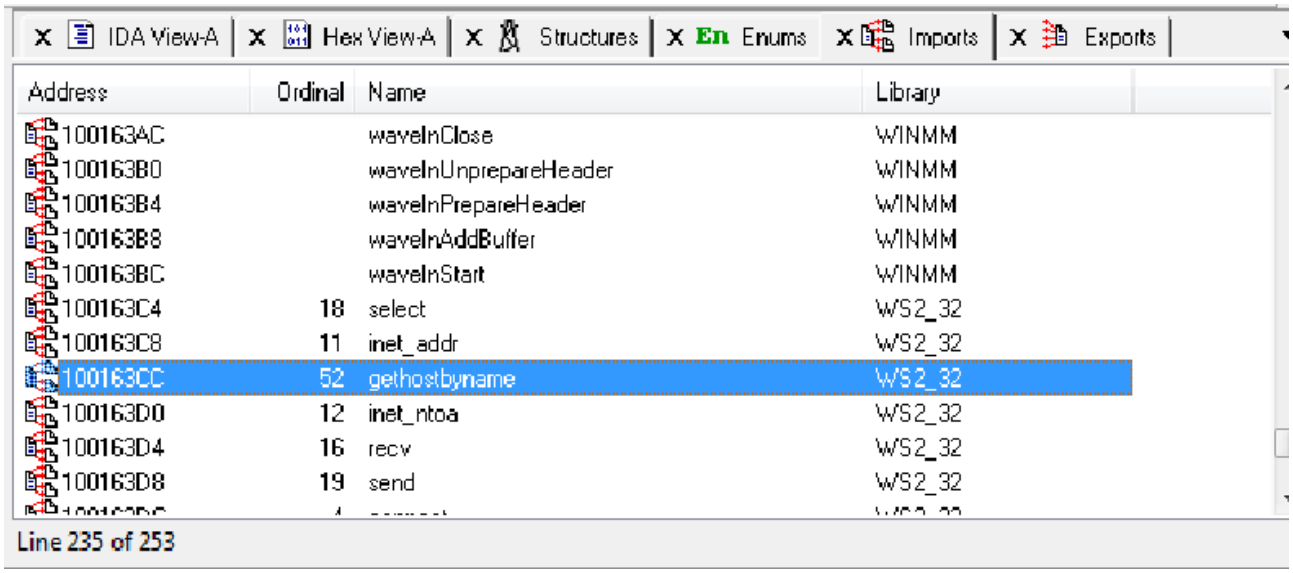
1. INDIVIDUARE INDIRIZZO DELLA FUNZIONE DLL MAIN

Rimanendo nella prima sezione del programma, IDA View A, possiamo andare a cercare l'indirizzo corrispondente alla funzione *dll main*, che in questo caso è allocata all'indirizzo 1000D02E.



2. INDIVIDUARE LA FUNZIONE "GETHOSTBYNAME"

Spostandoci nella sezione imports del programma IDA, possiamo andare a visualizzare tutte le funzioni importate dal malware. Con una semplice ricerca possiamo individuare la funzione *gethostbyname*, che si trova all'indirizzo 100163CC.



3. VARIBIALI LOCALI ALLA LOCAZIONE DI MEMORIA 0x10001656

In questo caso dobbiamo tornare nella sezione *IDA View A*, e scorrendo, o cercando la locazione di memoria 0x10001656, possiamo trovare tutte le variabili o parametri contenuti in quella locazione.

```
.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:
.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
```

Sapendo che le variabili locali hanno un offset negativo rispetto ad EBP, possiamo identificare 20 variabili.

4. PARAMETRI ALLA LOCAZIONE DI MEMORIA 0x10001656

Continuando ad analizzare la locazione di memoria 0x10001656, possiamo trovare un unico parametro *arg_0*, in quanto è l'unico ad avere un offset positivo rispetto ad EBP.

```
.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:
.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
```