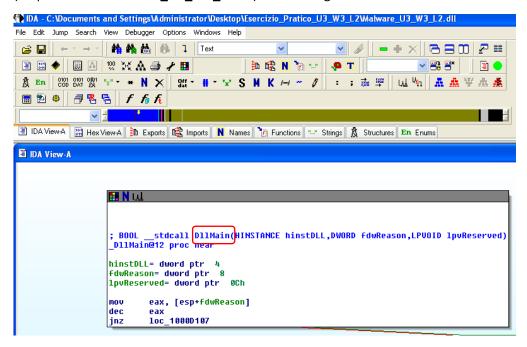
# Analisi statica avanzata con IDA

# Malware\_U3\_W3\_L2

### 1. Individuare l'indirizzo della funzione DLLMain (così com'è, in esadecimale)

Nel pannello centrale di IDA vengono mostrate le traduzioni in Assembly del codice macchina dell'eseguibile (in questo caso Malware\_U3\_W3\_L2.dll). Nel rettangolo rosso vediamo la funzione DllMain.

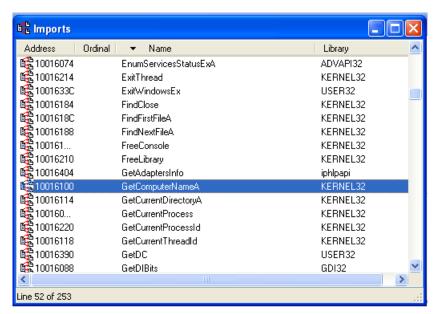


Premendo la barra spaziatrice si attiva la versione testuale dell'interfaccia grafica vista prima. Da qua possiamo vedere che la funzione **DllMain si trova all'indirizzo 1000D02E.** 

```
IDA View-A
        .text:1000D02E
        .text:1000D02E
         text:1000D02E
        _text:1000D02E; BOOL __stdcall DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPVOID lpvReserved)
        .text:1000D02E _D11Main@12
                                                          ; CODE XREF: DllEntryPoint+4Blp
                                    proc near
        .text:1000D02E
                                                          ; DATA XREF: sub_100110FF+2D40
        .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]
        .text:1000D032
                                    dec
                                            eax
        .text:1000D033
                                            loc_1000D107
                                    jnz
        .text:1000D039
                                            eax, [esp+hinstDLL]
                                    mov
        .text:1000D03D
                                    push
                                            ebx
                                            ds:hModule, eax
        .text:1000D03E
                                    mov
                                           eax, off_10019044
        .text:1000D043
                                    mnu
```

## 2. Dalla scheda "Imports" individuare la funzione "GetComputerName". Qual è l'indirizzo dell'import?

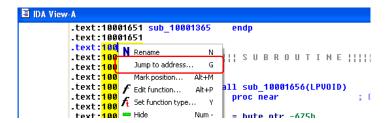
La scheda Imports mostra le funzioni importate dall'eseguibile. La prima colonna, Address, specifica l'indirizzo della funzione, che come evidenziato nell'immagine è **10016100**.



#### 3. Quante sono le variabili locali della funzione alla locazione di memoria 0x10001656?

#### 4. Quanti sono invece i parametri della funzione sopra?

Dall'interfaccia testuale col tasto destro selezionamo l'opzione "Jump to address.." per saltare all'indirizzo di memoria richiesto.



A questo indirizzo dall'interfaccia testuale di IDA possiamo vedere che viene chiamata la funzione DllMain.

```
.text:10001656 ; DWORD __stdcall sub_10001656(LPV0ID)
.text:10001656 sub_10001656 proc near ; DATA XREF: DllMain(x,x,x)+C8io
```

Nelle sezioni sottostanti vediamo in verde i parametri e le variabili della funzione. Con IDA sappiamo che la variabili hanno un offset negativo rispetto al registro ebp, mentre i parametri hanno un offset positivo.

L'offset, evidenziato nella figura sotto, è riportato in verde in coda al parametro / variabile.

```
.text:<mark>10001656</mark>
.text:10001656 var_675
.text:10001656 var_674
.text:10001656 hModule
                                                           byte ptr -675h
dword ptr -674h
dword ptr -670h
                                                           timeval ptr -66Ch
sockaddr ptr -664h
word ptr -654h
.text: 10001656 timeout
.text:<mark>10001656</mark> name
.text:<mark>10001656</mark> var_654
                                                           in_addr ptr -650h
.text:<mark>10001656</mark> in
                                                          byte ptr -644h
byte ptr -63Fh
.text:<mark>10001656</mark> Parameter
.text:<mark>10001656</mark> CommandLine
text:10001656 Data
.text:10001656 var_544
.text:10001656 var_500
.text:10001656 var_500
.text:10001656 var_4FC
                                                       = byte ptr -638h
                                                           dword ptr -544h
                                                           dword ptr -50Ch
                                                          dword ptr -500h
dword ptr -4FCh
fd_set ptr -4BCh
HKEY__ ptr -3B8h
dword ptr -3B0h
.text:10001656 readfds
.text:10001656 phkResult
.text:<mark>10001656</mark> var_3B0
.text:10001656 var_1A4
.text:10001656 var_194
                                                           dword ptr -1A4h
                                                           dword ptr -194h
.text: 10001656 WSAData
                                                           WSAData ptr -190h
.text:<mark>10001656</mark> arg_0
                                                           dword ptr 4
```

Abbiamo un totale di 21 voci, di cui 20 hanno offset negativo rispetto a ebp, e sono quindi variabili della funzione, mentre 1, arg\_0, ha un offset positivo. arg\_0 è quindi l'unico parametro della funzione.