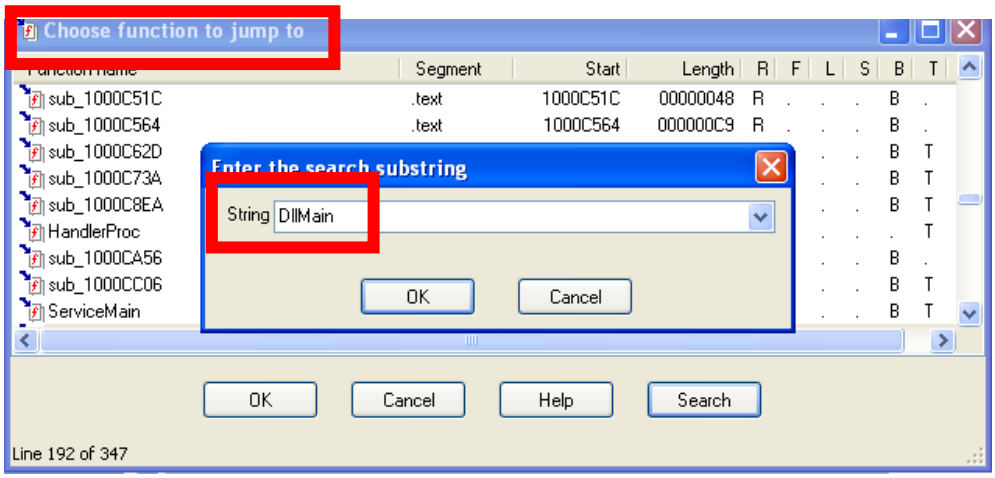
ANALISI STATICA AVANZATA

DATO IL FILE.DLL MALWARE_U3_W3_L2:

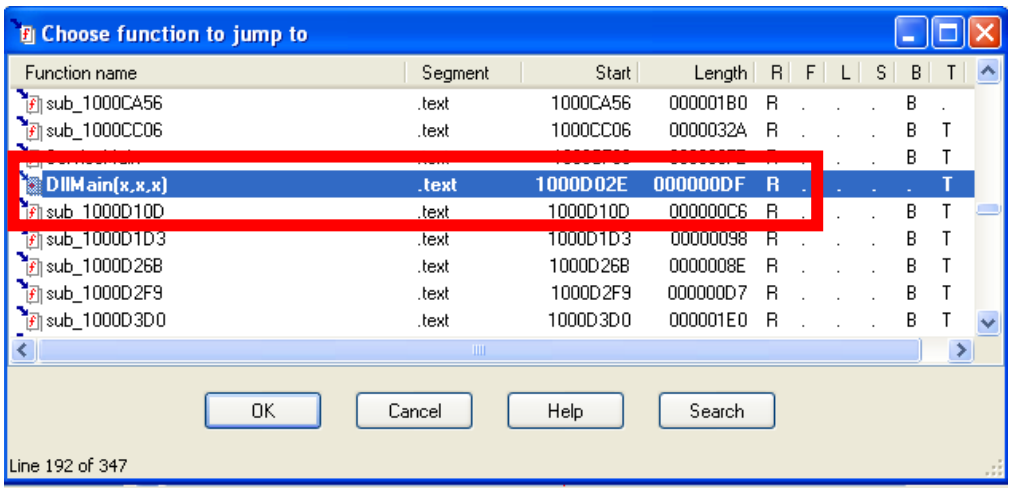
- **Individuare l'indirizzo della funzione DLLMain in esadecimale**
- **Individuare la funzione gethostbyname e l'indirizzo dell'import**
- **Quante sono le variabili locali della funzione alla locazione di memoria 0x10001656?**
- **Quanti sono i parametri della funzione sopra?**
- **Considerazioni macro livello sul malware**

TASK 1: Individuare l'indirizzo della funzione DLLMain in esadecimale

Ho avviato IDA Pro, e nel menu JUMP ho selezioanto Jump to function, inserendo la stringa "DllMain" (funzione oggetto d'interesse).

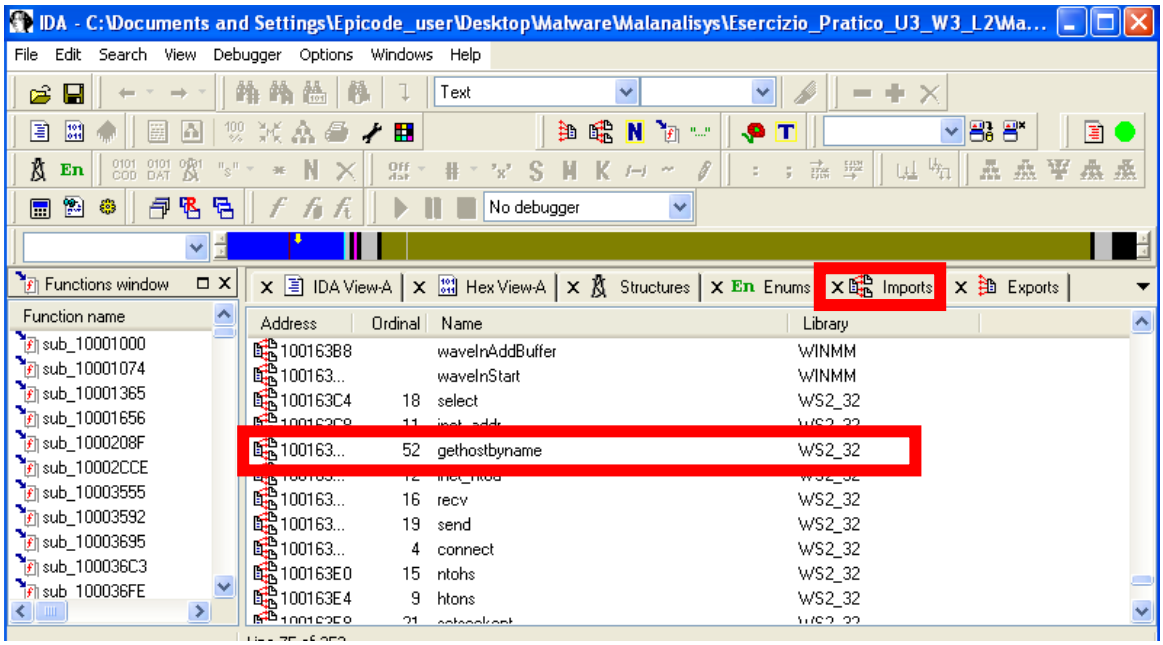


L'indirizzo di memoria associato alla funzione **DllMain** è **1000D02E**

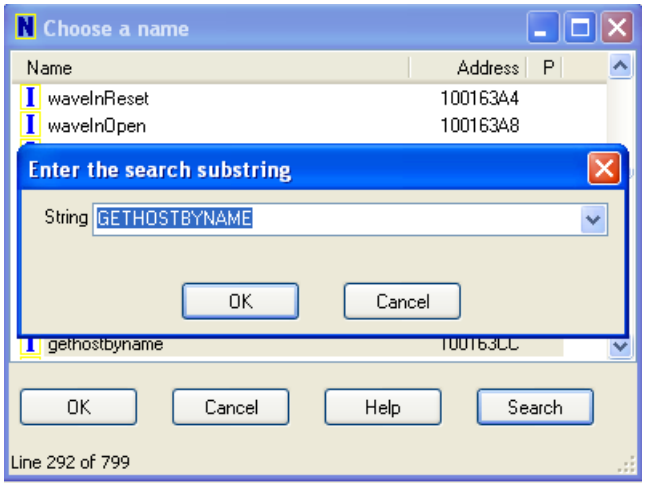


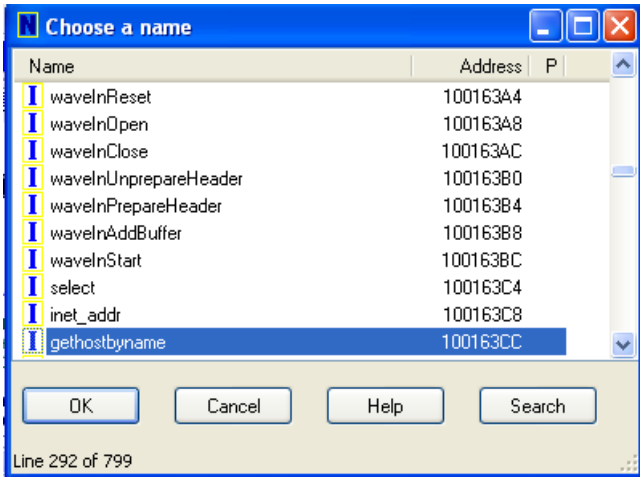
TASK 2 = Individuare la funzione gethostbyname e l'indirizzo dell'import

Dalla schermata "**IMPORTS**" ho rintracciato la funzione **GETHOSTBYNAME**, presente all'indirizzo **1001063CC**.



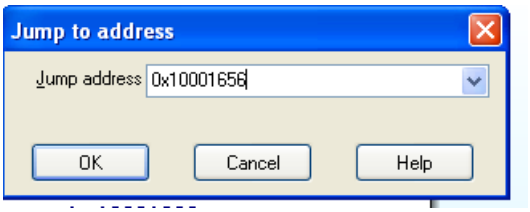
Stesso procedimento effettuato con **JUMP** (**gethostbyname**)





TASK 3 E TASK 4 = Quante sono le variabili locali della funzione alla locazione di memoria 0x10001656? E quanti sono i parametri?

JUMP TO ADDRESS → 0x10001656




```
.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 hLibModule      = dword ptr -670h
.text:10001656 timeout         = timeval ptr -66Ch
.text:10001656 name            = sockaddr ptr -664h
.text:10001656 var_654         = word ptr -654h
.text:10001656 Dst             = dword ptr -650h
.text:10001656 Parameter       = byte ptr -644h
.text:10001656 var_640         = byte ptr -640h
.text:10001656 CommandLine     = byte ptr -63Fh

.text:10001656 Source          = byte ptr -63Dh
.text:10001656 Data            = byte ptr -638h
.text:10001656 var_637         = byte ptr -637h
.text:10001656 var_544         = dword ptr -544h
.text:10001656 var_50C         = dword ptr -50Ch
.text:10001656 var_500         = dword ptr -500h
.text:10001656 Buf2            = byte ptr -4FCh
.text:10001656 readfds         = fd_set ptr -4BCh
.text:10001656 phkResult       = byte 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
```

Il risultato della ricerca JUMP TO ADDRESS ci restituisce la funzione di tipo **subroutine sub_10001656**, comosta da **23 variabili locali** e **1 parametro.**

Il tool **IDA differenzia variabili e parametri utilizzando come riferimento l'offset** (differenza rispetto ad un valore di riferimento) rispetto al puntatore EBP:

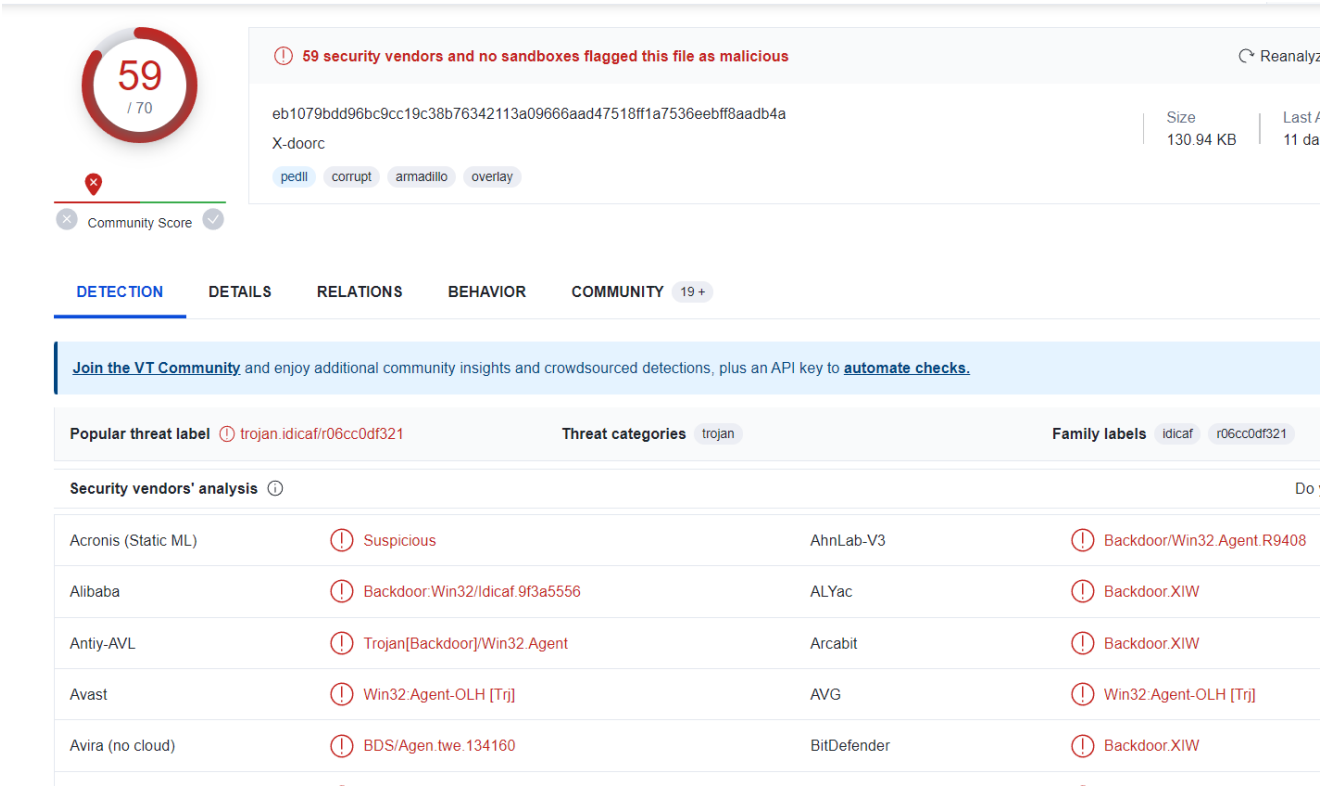
- Le **variabili** sono ad un **offset negativo** rispetto al registro EBP
- I **parametri** si trovano ad un **offset positivo** rispetto al registro EBP

TASK 5: Considerazioni macro livello sul malware

VIRUSTOTAL

Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community.

1A9FD80174AAFECD9A52FD908CB82637



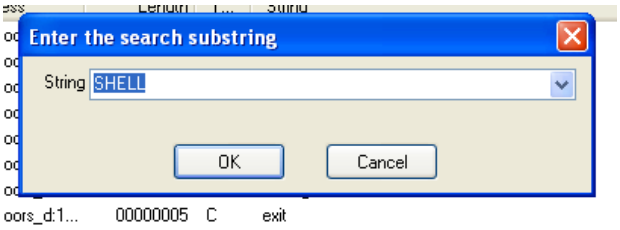
il malware ha lo scopo di ottenere la **persistenza** dentro il sistema della macchina vittima, aggiungendo sé stesso alle entry dei programmi che devono essere eseguiti all'avvio del PC, in modo tale da essere eseguito in maniera automatica e permanente senza alcun intervento da parte dell'utente. Per far ciò, il malware richiede l'accesso e la modifica ad una chiave di registro tramite due chiamate di funzione principali:

La funzione **RegOpenKeyEx** permette di aprire una chiave di registro al fine di modificarla. Essa accetta come parametri, tra gli altri, la chiave da aprire.

La funzione **RegSetValueEx** permette invece di aggiungere un nuovo valore all'interno del registro e di settare i rispettivi dati. Accetta come parametri la chiave, la sottochiave e il dato da inserire.

```
push    esi
push    eax             ; phkResult
xor     esi, esi
push    0F003Fh         ; samDesired
push    esi             ; ulOptions
push    offset aSoftwareMicros ; "SOFTWARE\\Microsoft\\Windows\\CurrentVersi"...
push    esi             ; hKey
call    ds:RegOpenKeyExA
test    eax, eax
jnz     short loc_1000568F
lea     eax, [ebp+Data]
push    4               ; cbData
push    eax             ; lpData
push    4               ; dwType
```

```
push    esi             ; Reserved
push    [ebp+lpValueName] ; lpValueName
push    [ebp+hKey]      ; hKey
call    ds:RegSetValueExA
test    eax, eax
jnz     short loc_1000568F
```




```
    db 0Dh,0Ah
    db 'Machine UpTime  [%-.2d Days %-.2d Hours %-.2d Minutes %-.2d
    db 'ds]',0Dh,0Ah
    db 'Machine IdleTime [%-.2d Days %-.2d Hours %-.2d Minutes %-.2
    db 'nds]',0Dh,0Ah
    db 0Dh,0Ah
    db 'Encrypt Magic Number For This Remote Shell Session [0x%02x]
    db 0Dh,0Ah,0
0095C5C[]
:                               ; DATA XREF: sub_1000FF58+4B↑o
                                ; sub_1000FF58+3E1↑o
    dw 3Eh, 0
    align 400h
    ends
```

```
"..." xdoors_d:1...   00000005  C   quit
"..." xdoors_d:1...   00000011  C   \\command.exe /c
"..." xdoors_d:1...   0000000D  C   \\cmd.exe /c
"..." xdoors_d:1...   00000118  C   Hi,Master [%d/%d/%d %d:%d:%d]\r\nWelCome Back...Are You Enjoying To...
```