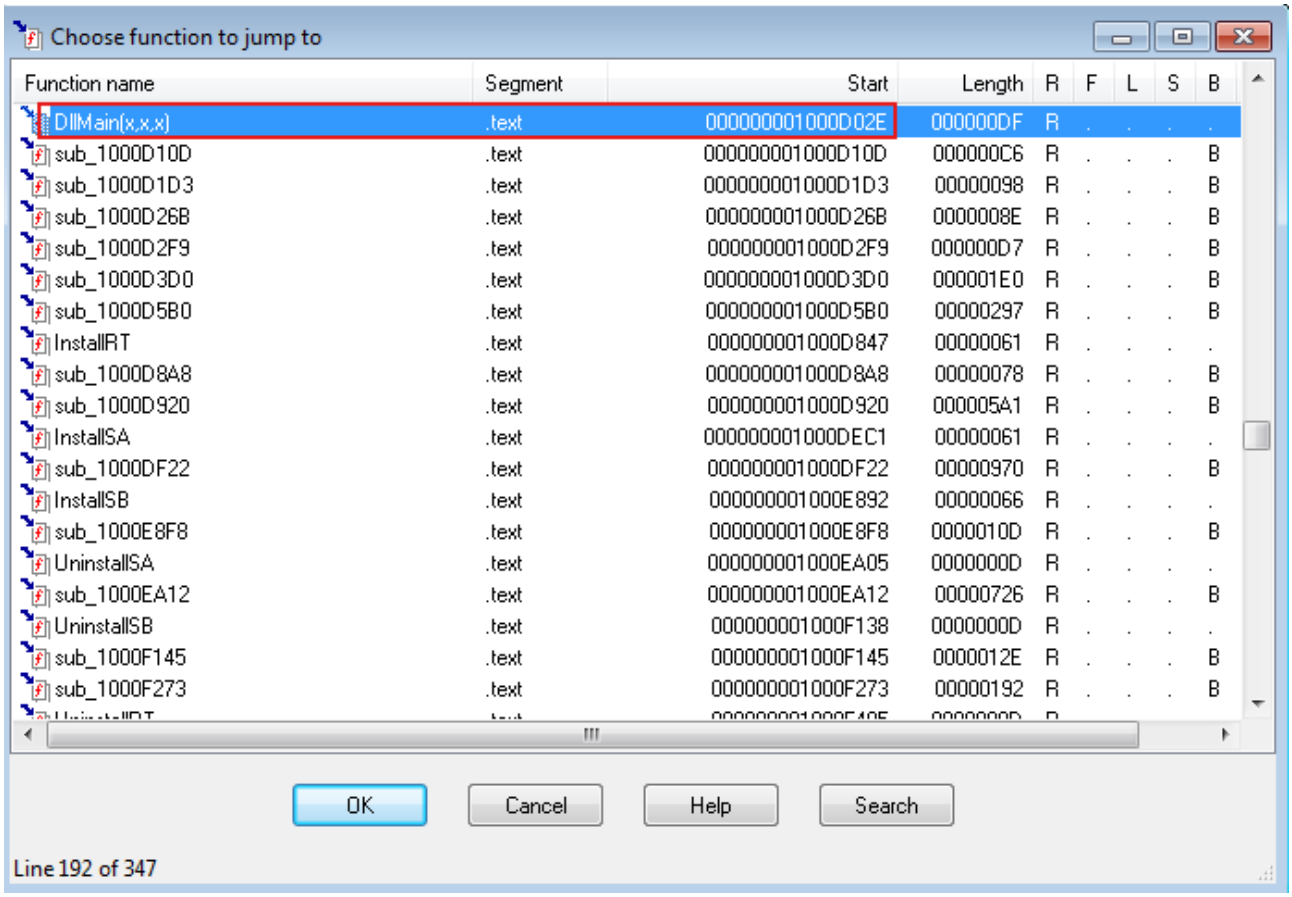
CONSEGNA U3 S11 L2

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

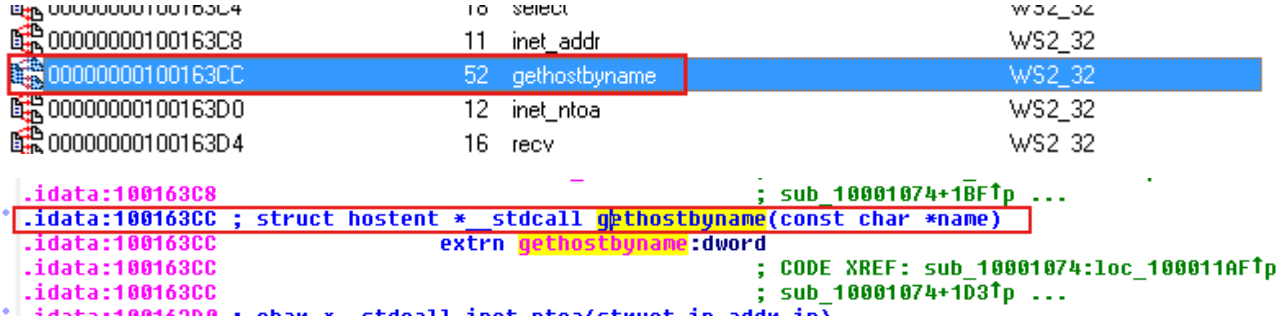
L'indirizzo della funzione è **0x000000001000D02E.**




```
.text:1000D02E
.text:1000D02E ; BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
.text:1000D02E _DllMain@12     proc near               ; CODE XREF: DllEntryPoint+4B↓p
.text:1000D02E                                         ; DATA XREF: sub_100110FF+2D↓o
.text:1000D02E
.text:1000D02E hinstDLL        = dword ptr  4
.text:1000D02E fdwReason       = dword ptr  8
.text:1000D02E lpvReserved     = dword ptr  0Ch
.text:1000D02E
```

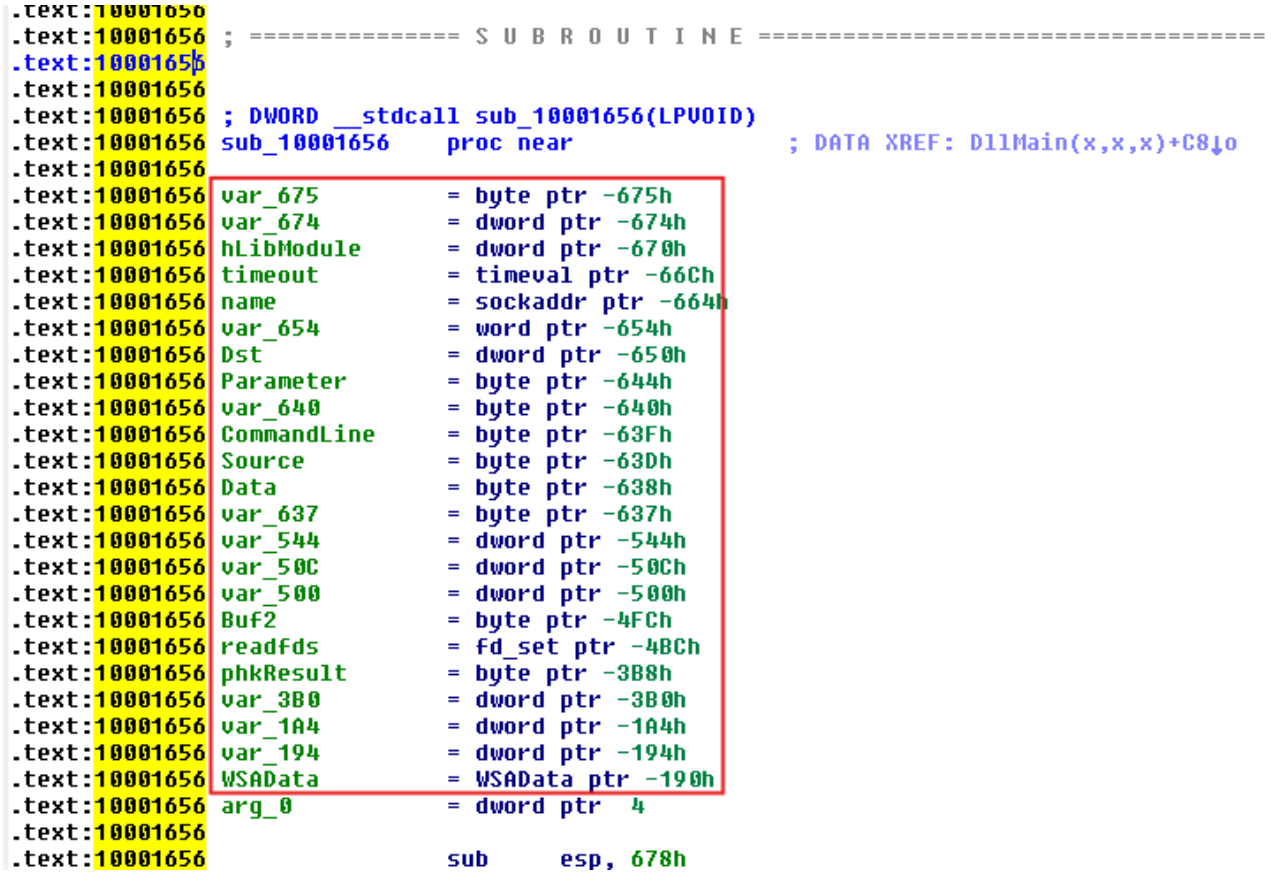
- **Dalla scheda «imports» individuare la funzione «gethostbyname». Qual è l'indirizzo dell'import? Cosa fa la funzione?**

La funzione **gethostbyname()** restituisce l'indirizzo IP per un determinato nome di dominio/host. Il suo indirizzo è **0x00000000100163CC**.



- **Quante sono le variabili locali della funzione alla locazione di memoria 0x10001656?**

Le variabili locali sono **23**. Si identificano mediante **l'offset negativo (es. -654h).**



- **Quanti sono, invece, i parametri della funzione sopra?**

Il parametro che viene passato alla funzione è unico ed è **arg_0**.

```
.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 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
.text:10001656
```

- **Inserire altre considerazioni macro livello sul malware (comportamento)**

Dalle varie funzioni che va a richiamare il software malevolo, ho ipotizzato che potrebbe trattarsi di una **backdoor** o un software per eseguire un **C2 Server (Command & Control Server)** magari per creare macchine zombie e sfruttarle per delle botnet. Il programma può avere accesso alla possibilità di immettere comandi nella macchina infetta usando mouse e tastiera. Infatti il software malevolo prima di tutto apre una comunicazione verso la rete. Poi va a richiamare delle funzioni che hanno a che fare con l'inserimento input utente e le funzionalità di desktop remoto. Inoltre fa uso delle funzioni **send** e **recv** , probabilmente per inviare verso l'esterno i dati raccolti.