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. Cosa fa la funzione?**
- **Quante sono le variabili locali della funzione alla locazione di memoria0x10001656?**
- **Quanti sono i parametri della funzione sopra?**
- **Considerazioni macro livello sul malware**

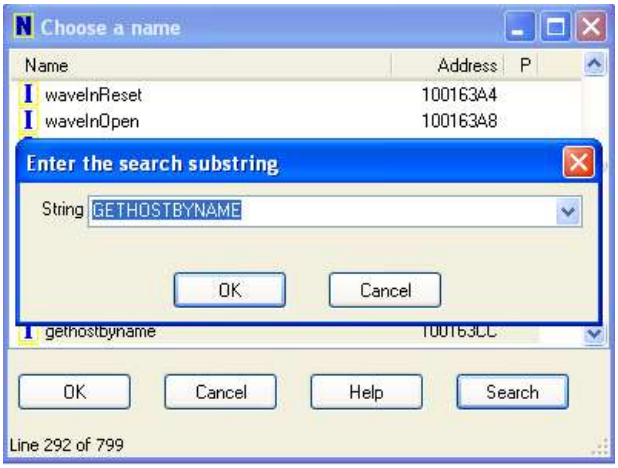
TASK 1: Individuare l'indirizzo della funzione DLLMain in esadecimale

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

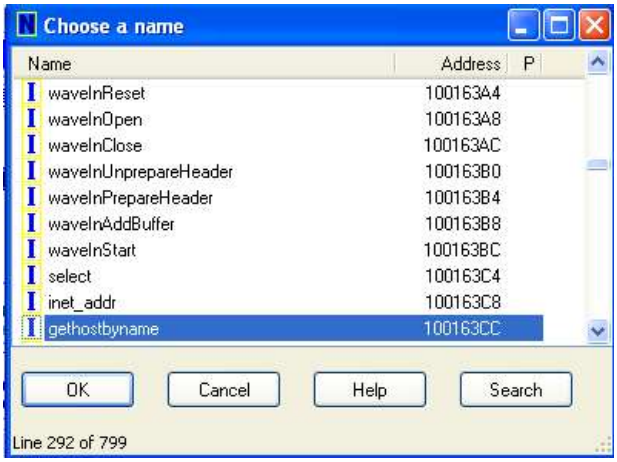
Dalla schermata "**IMPORTS**" ho rintracciato la funzione **GETHOSTBYNAME**, presente all'indirizzo
1001063CC.



Stesso procedmentoeffettuato con **JUMP**(**gethostbyname**)

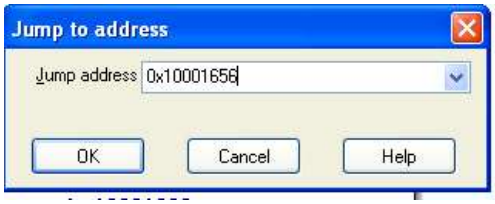
La funzione gethostbyname è una funzione di programmazione di socket che viene utilizzata per ottenere informazioni associate a un nome host. Questa funzione è spesso utilizzata in programmi che coinvolgono la comunicazione di rete, come quelli che creano connessioni TCP/IP.

In particolare, la funzione gethostbyname restituisce un oggetto di tipo hostent che contiene informazioni sul nome host fornito come parametro. Le informazioni includono l'indirizzo IP associato al nome host.



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

JUMP TO ADDRESS → 0x10001656

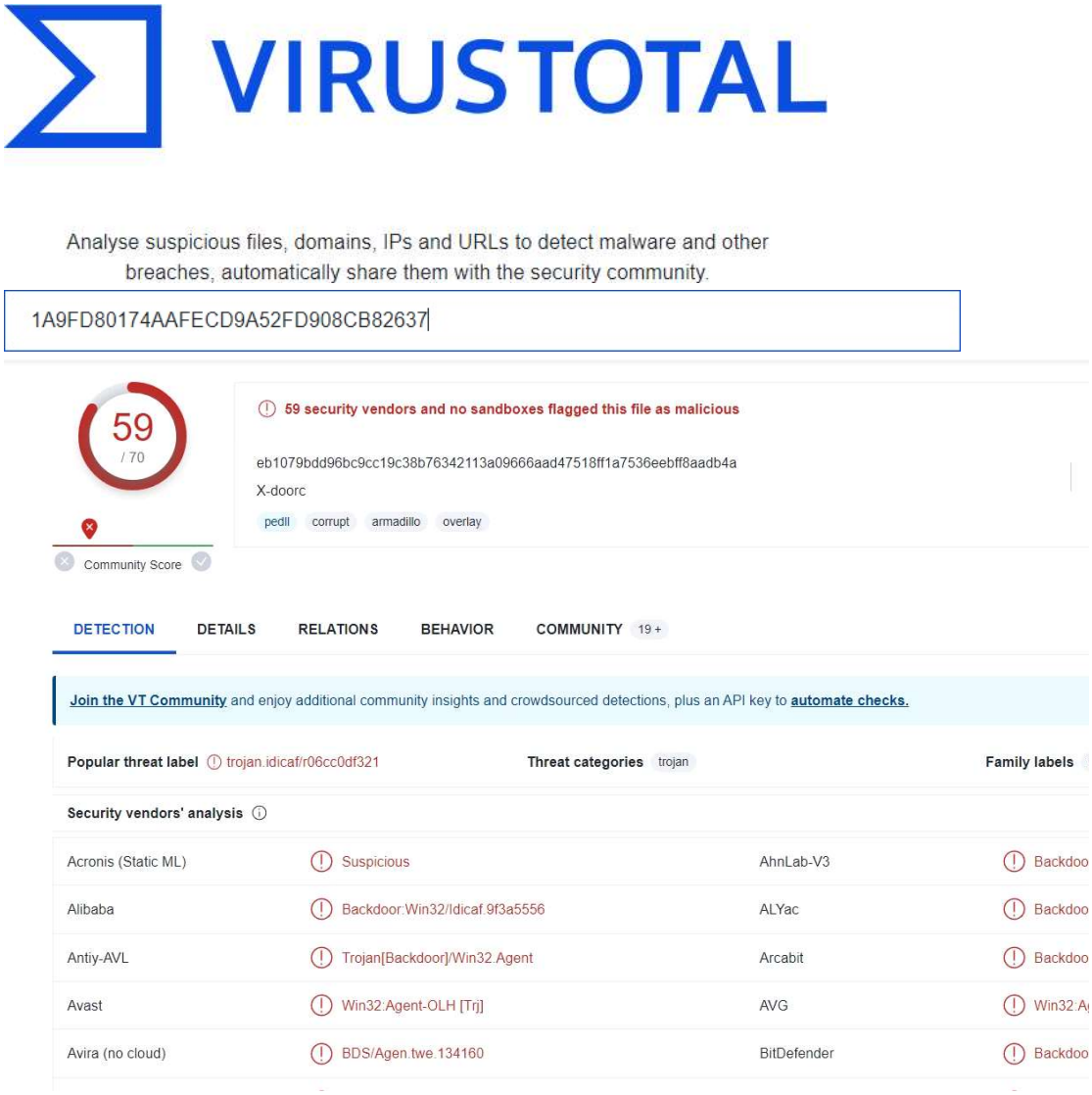



```
10001656 var_675         = byte ptr -675h
10001656 var_674         = dword ptr -674h
10001656 hModule         = dword ptr -670h
10001656 timeout         = timeval ptr -66Ch
10001656 name            = sockaddr ptr -664h
10001656 var_654         = word ptr -654h
10001656 in              = in_addr ptr -650h
10001656 Parameter       = byte ptr -644h
10001656 CommandLine     = byte ptr -63Fh
10001656 Data            = byte ptr -638h
10001656 var_544         = dword ptr -544h
10001656 var_50C         = dword ptr -50Ch
10001656 var_500         = dword ptr -500h
10001656 var_4FC         = dword ptr -4FCh
10001656 readfds         = fd_set ptr -4BCh
10001656 phkResult       = HKEY__ ptr -3B8h
10001656 var_3B0         = dword ptr -3B0h
10001656 var_1A4         = dword ptr -1A4h
10001656 var_194         = dword ptr -194h
10001656 WSAData         = WSAData ptr -190h
10001656 arg_0           = dword ptr  4
```

----------**Parametro**

Il risultato della ricerca JUMP TO ADDRESS ci restituisce la funzione di tipo **subroutine sub_10001656**,composta da **20 variabili locali** e **1 parametro.**

TASK 5: Considerazioni macro livello sul malware



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 eseguitoin 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 accettacome 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.

Carica una shell, ergo potrebbe essere una backdoor (anche perché ha ottenuto lapersistenza)