

S11-L2

Traccia

Lo scopo dell'esercizio di oggi è di acquisire esperienza con IDA, un tool fondamentale per l'analisi statica.

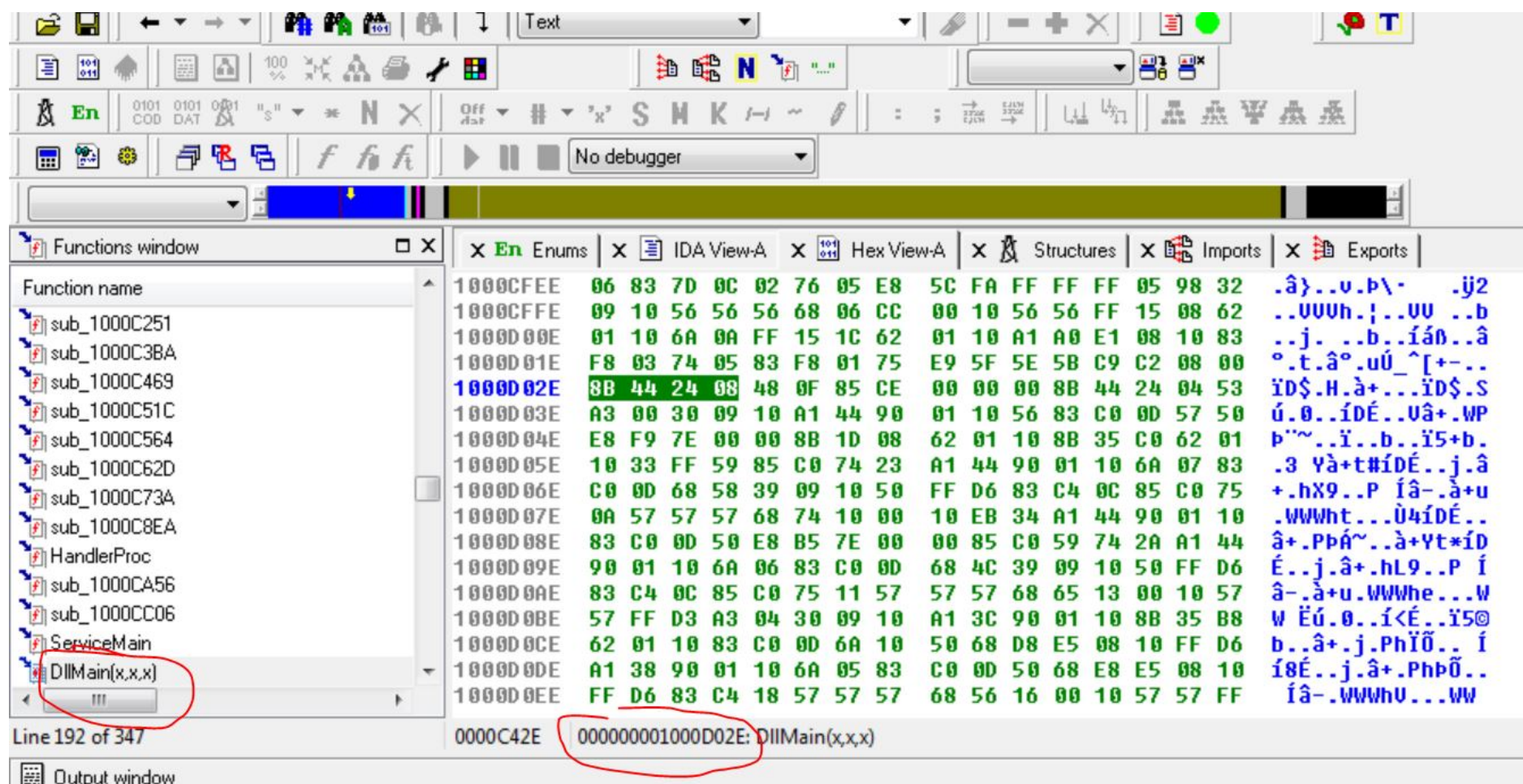
A tal proposito, con riferimento al malware chiamato «Malware_U3_W3_L2» presente all'interno della cartella «Esercizio_Pratico_U3_W3_L2» sul Desktop della macchina virtuale dedicata all'analisi dei malware, rispondere ai seguenti quesiti, utilizzando IDA Pro.

1. Individuare l'indirizzo della funzione DLLMain (così com'è, in esadecimale)
2. Dalla scheda «imports» individuare la funzione «gethostbyname». Qual è l'indirizzo dell'import? Cosa fa la funzione?
3. Quante sono le variabili locali della funzione alla locazione di memoria 0x10001656?
4. Quanti sono, invece, i parametri della funzione sopra?
5. Inserire altre considerazioni macro livello sul malware (comportamento)

Punto 1

Cliccando su Functions window possiamo lanciare lo short-cut alt+t e digitare ddllmain per identificare immediatamente la funzione richiesta.

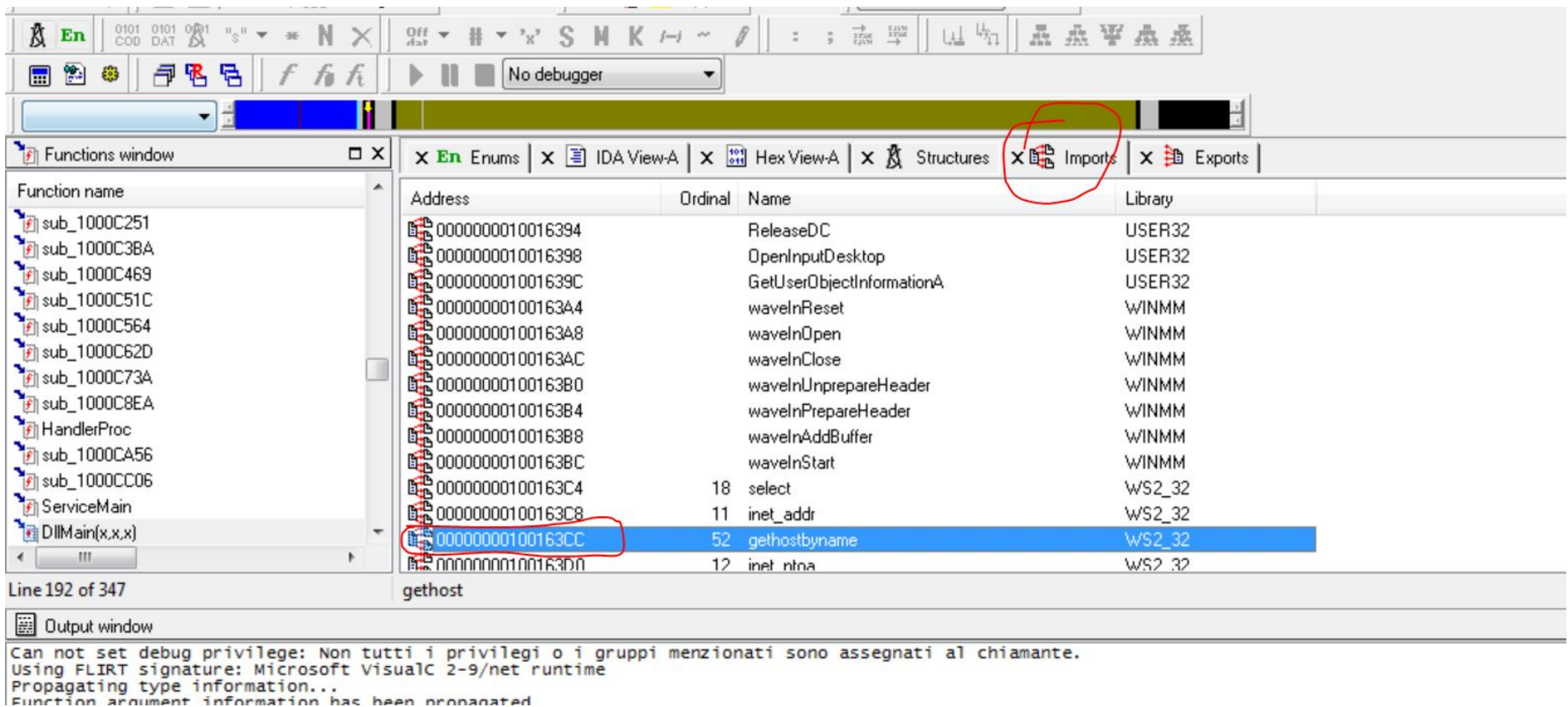
Da qui andiamo sul sotto-menù «Enums», ricicchiamo la funzione e ci verrà evidenziato l'indirizzo sulla sinistra, mentre in verde vediamo la codifica esadecimale.



Punto 2

Andiamo su «Imports» come evidenziato nell'immagine sotto e digitiamo «gethostbyname».

Individuato il nostro obiettivo, possiamo vedere l'indirizzo sulla sua sinistra, cerchiato nell'immagine.



Possiamo informarci su cosa faccia gethostbyname andando su [learn.microsoft](https://learn.microsoft.com/en-us/windows/win32/api/ws2_32/nf-ws2_32-gethostbyname) ove troveremo:

"La funzione

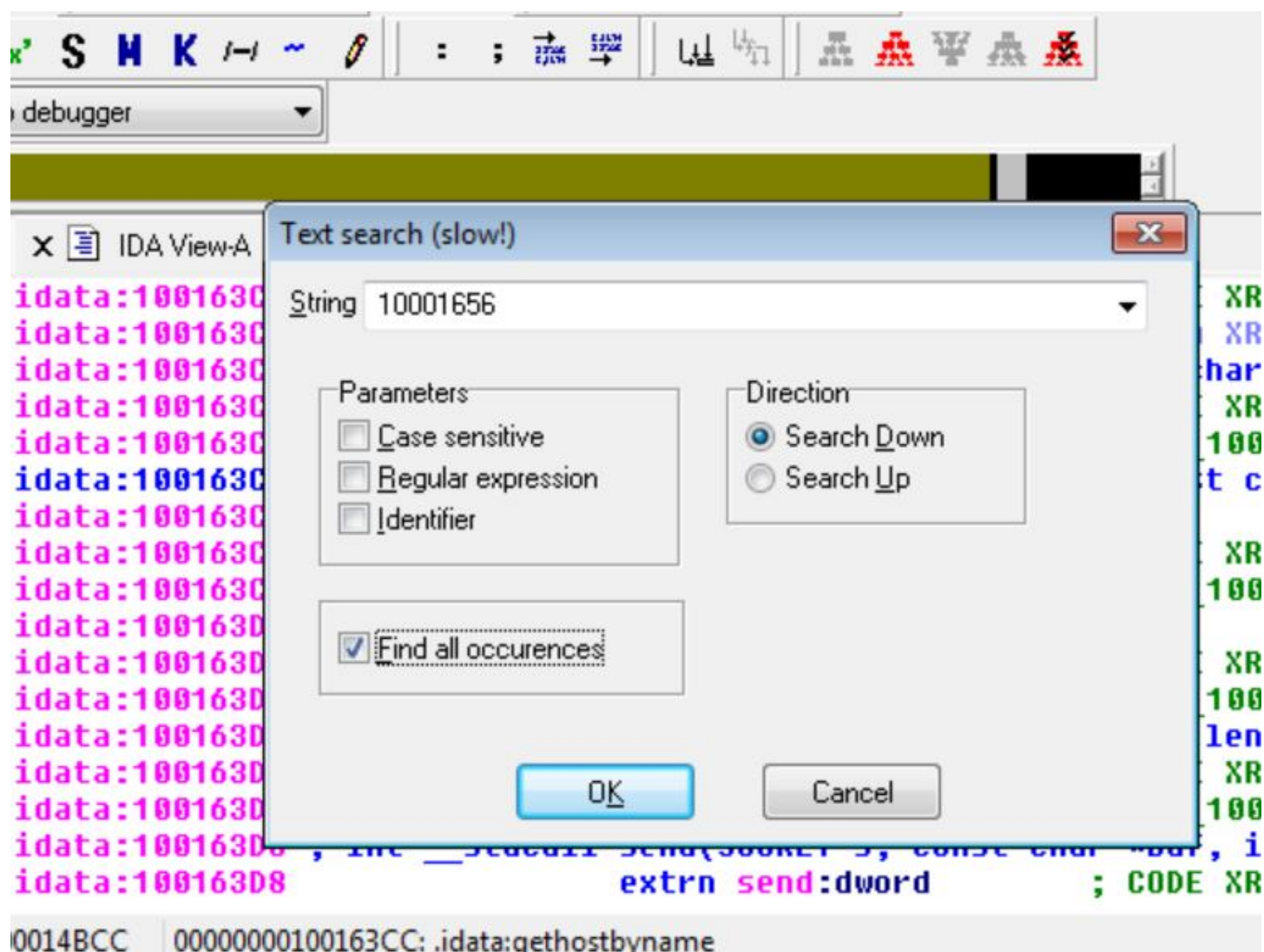
gethostbyname recupera le informazioni host corrispondenti a un nome host da un database host.

Se non si verifica alcun errore,

gethostbyname restituisce un puntatore alla struttura hostent descritta in precedenza. In caso contrario, restituisce un puntatore **Null** e un numero di errore specifico può essere recuperato chiamando WSAGetLastError."

Punto 3 e 4

Per iniziare andiamo ad individuare l'indirizzo di memoria tramite il tasto di ricerca (nell'immagine ho lasciato uno spazio prima dell'uno, quindi non trovava niente finchè non lo ho rimosso LOL).



Al termine della ricerca ci compare il codice da analizzare:


```

10001656 | X Occurrences of: 10001656 | X Hex View-A | X St
source= byte ptr -630h
Data= byte ptr -638h
var_637= byte ptr -637h
var_544= dword ptr -544h
var_50C= dword ptr -50Ch
var_500= dword ptr -500h
Buf2= byte ptr -4FCh
readfds= fd_set ptr -4BCh
phkResult= byte ptr -3B8h
var_3B0= dword ptr -3B0h
var_1A4= dword ptr -1A4h
var_194= dword ptr -194h
WSAData= WSAData ptr -190h
arg_0= dword ptr 4

sub     esp, 678h
push    ebx
push    ebp

```

0000000010001656: sub_10001656

Sapendo che:

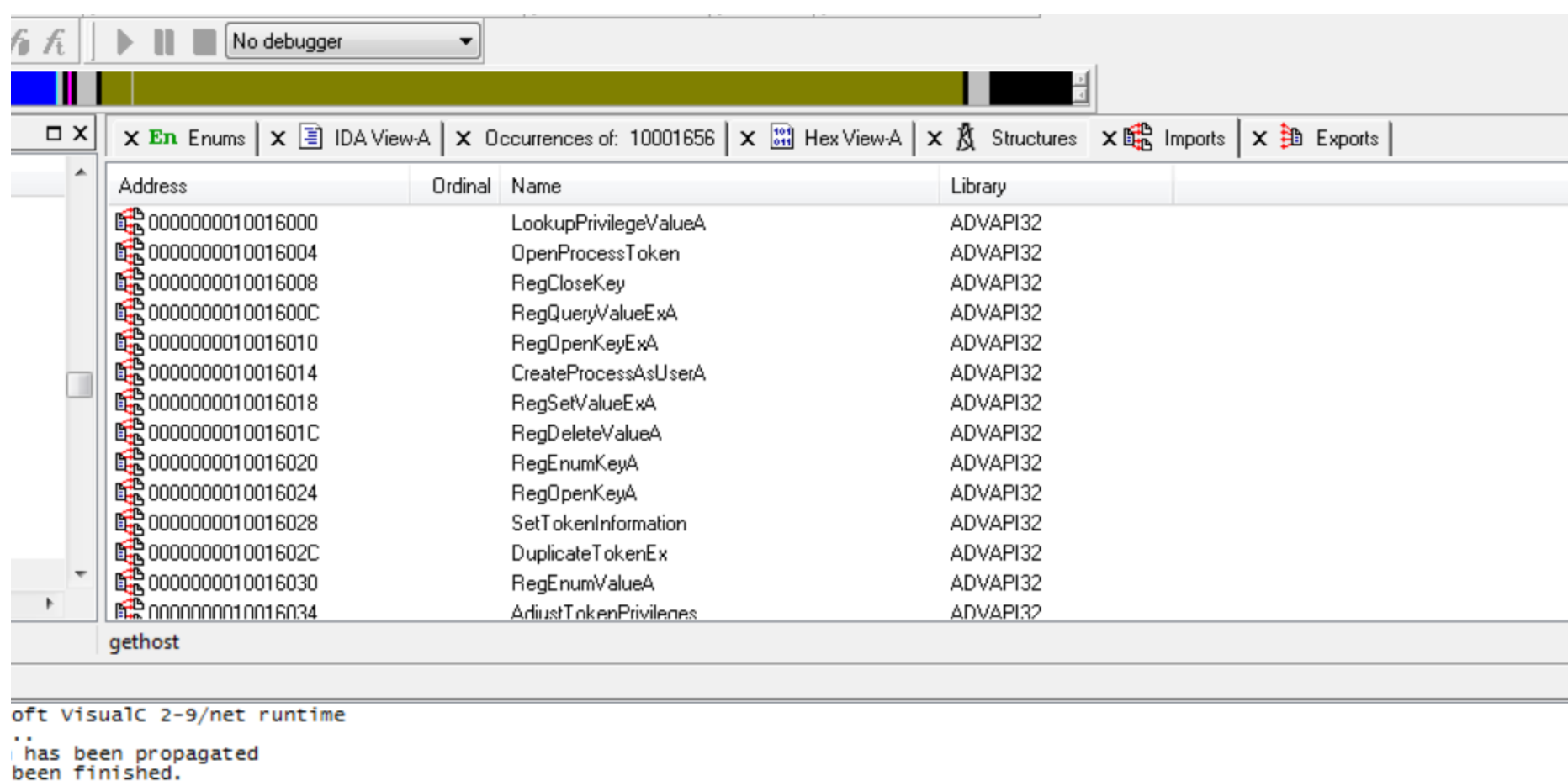
- Le variabili sono ad un offset negativo rispetto al registro EBP
- I parametri si trovano ad un offset positivo rispetto ad EBP

E che: con offset si intende la differenza rispetto ad un valore di riferimento; vediamo come le variabili sono 23, mentre il parametro è 1 (cerchiato).

Punto 5

Da un'analisi macro possiamo dedurre diverse cose sul comportamento di questo malware.

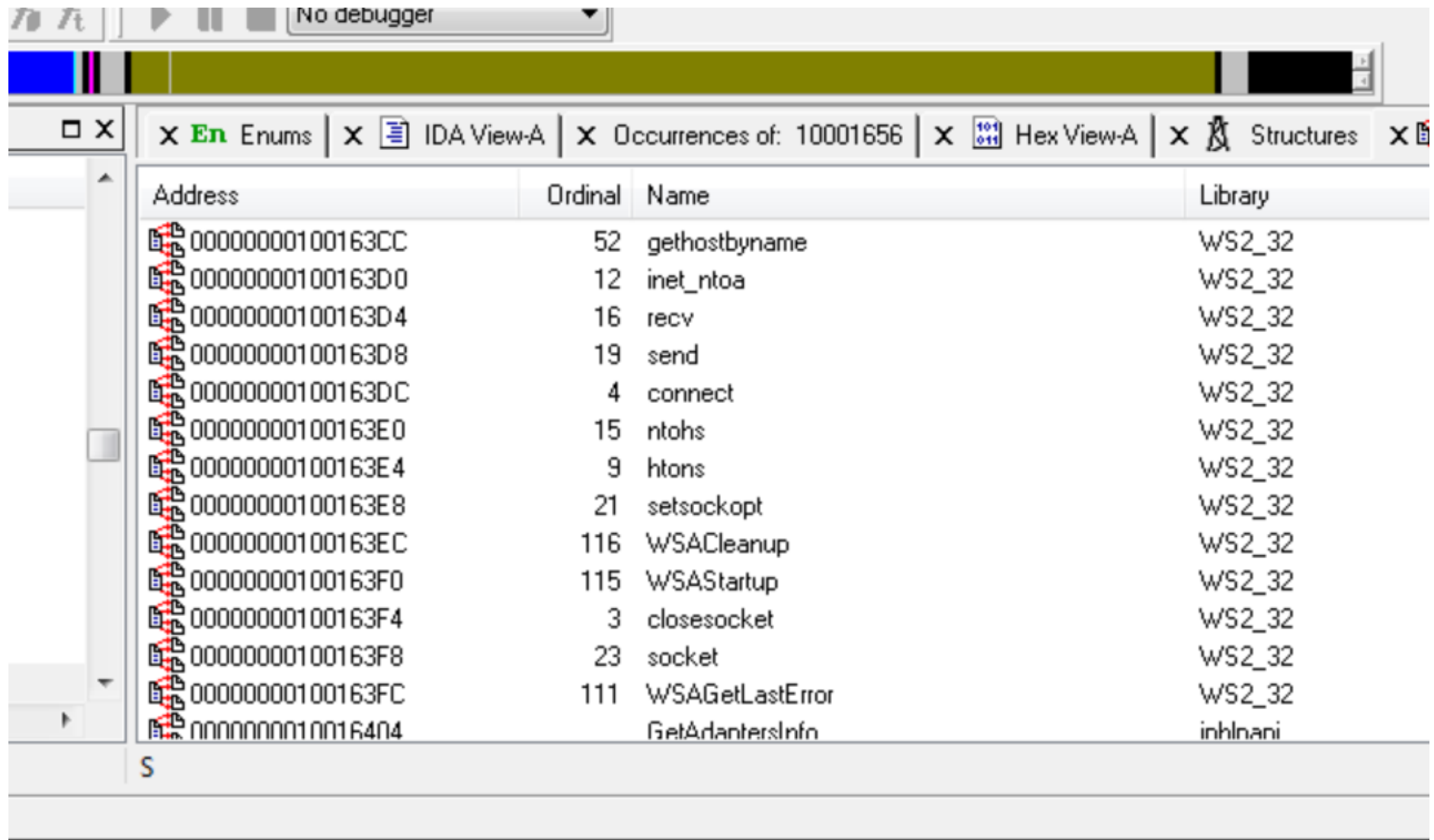
Iniziamo con il vedere, nell'immagine sotto, l'import di librerie aventi come fine la modifica di alcune key del registro.



Altre librerie hanno, invece, il fine ultimo di prendere informazioni e manipolare file.

00000000100160FC	GlobalMemoryStatus	KERNEL32
0000000010016100	GetComputerNameA	KERNEL32
0000000010016104	CopyFileA	KERNEL32
0000000010016108	MoveFileExA	KERNEL32
000000001001610C	GetModuleFileNameA	KERNEL32

Altre ancora, invece, permettono l'instaurazione di connessioni.



The screenshot shows the IDA Pro interface with the 'Enums' window open. The window title is 'X En Enums'. The main pane displays a list of functions with their addresses, ordinals, names, and libraries. The functions are all from the 'WS2_32' library, except for 'GetAdaptersInfo' which is from 'iphlpapi'. The list includes functions like 'gethostbyname', 'inet_ntoa', 'recv', 'send', 'connect', 'ntohs', 'htons', 'setsockopt', 'WSACleanup', 'WSAStartup', 'closesocket', 'socket', 'WSAGetLastError', and 'GetAdaptersInfo'.

Address	Ordinal	Name	Library
00000000100163CC	52	gethostbyname	WS2_32
00000000100163D0	12	inet_ntoa	WS2_32
00000000100163D4	16	recv	WS2_32
00000000100163D8	19	send	WS2_32
00000000100163DC	4	connect	WS2_32
00000000100163E0	15	ntohs	WS2_32
00000000100163E4	9	htons	WS2_32
00000000100163E8	21	setsockopt	WS2_32
00000000100163EC	116	WSACleanup	WS2_32
00000000100163F0	115	WSAStartup	WS2_32
00000000100163F4	3	closesocket	WS2_32
00000000100163F8	23	socket	WS2_32
00000000100163FC	111	WSAGetLastError	WS2_32
0000000010016404		GetAdaptersInfo	iphlpapi

Probabilmente questo malware agisce come una backdoor.