

Esercizio Analisi statica | Marco Malizia - DataShields

Traccia:

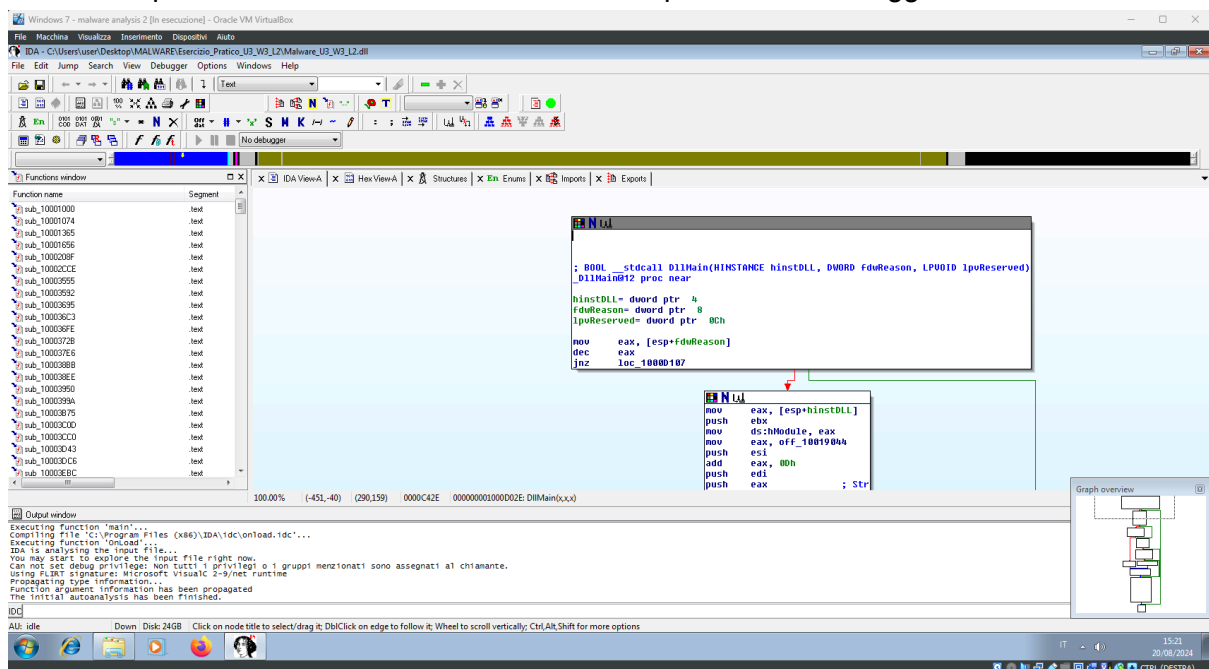
Lo scopo dell'esercizio di oggi è di acquisire esperienza con IDA, un tool fondamentale per l'analisi statica. A tal proposito, con riferimento al malware chiamato «Malware_U3_W3_L2» presente all'interno della cartella «Esercizio_Pratico_U3_W3_L2» sul Desktop della macchina virtuale dedicata all'analisi 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)

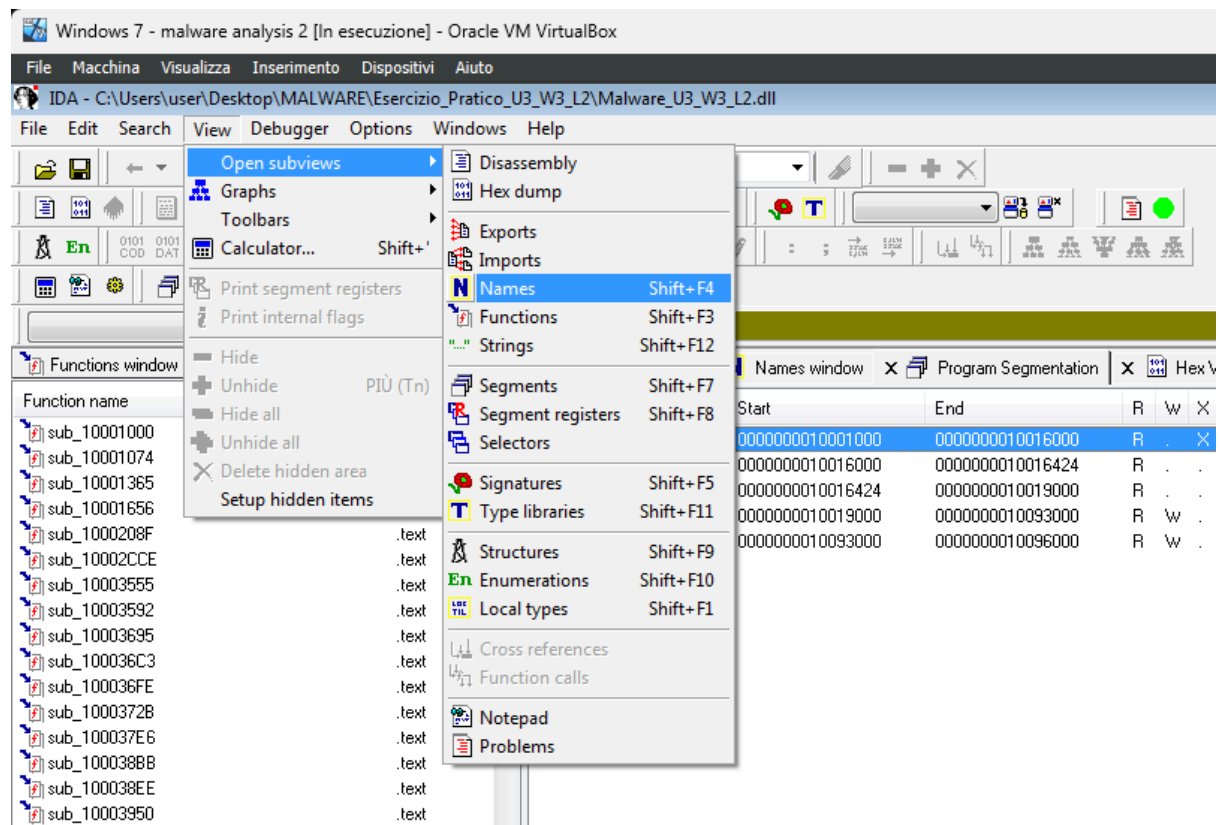
Svolgimento:

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

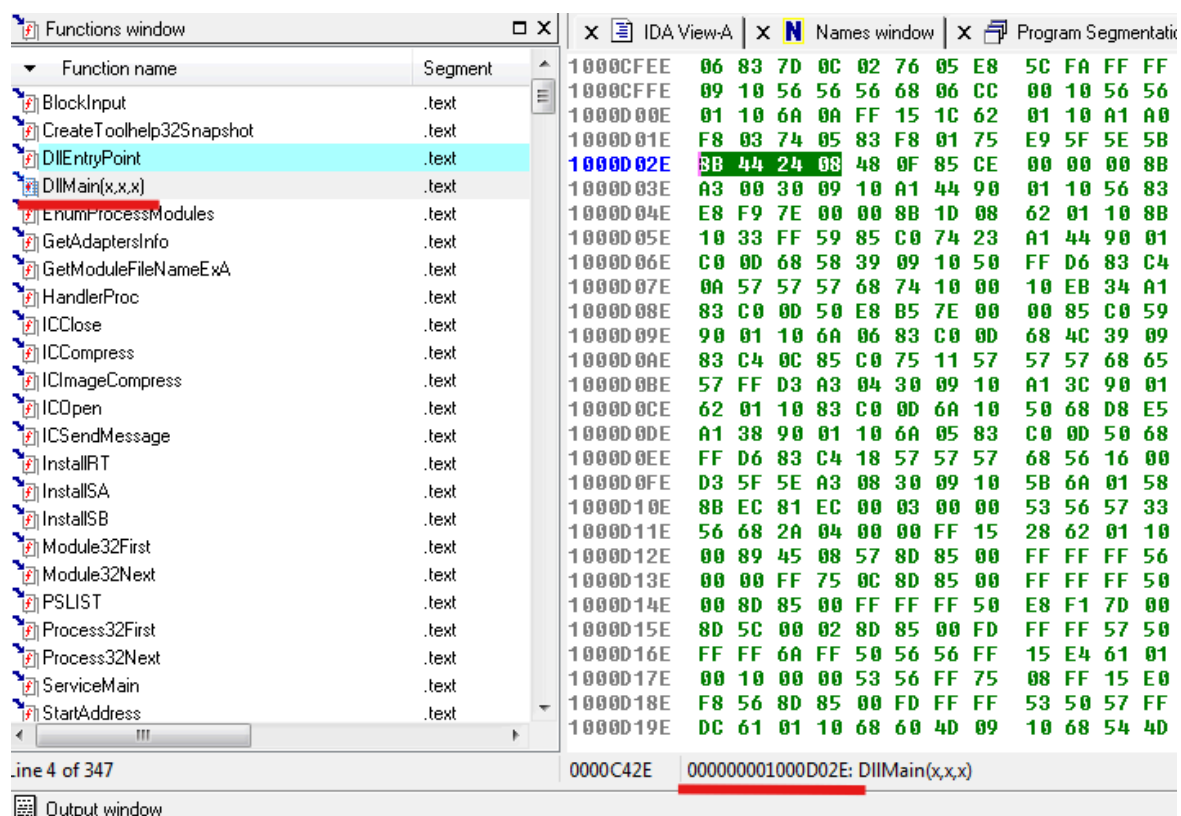
Una volta aperto il software IDA Pro, andiamo ad aprire il Malware oggetto di analisi.



Una volta aperto il file impostiamo la visualizzazione così da poter visualizzare i codici esadecimali delle funzioni.

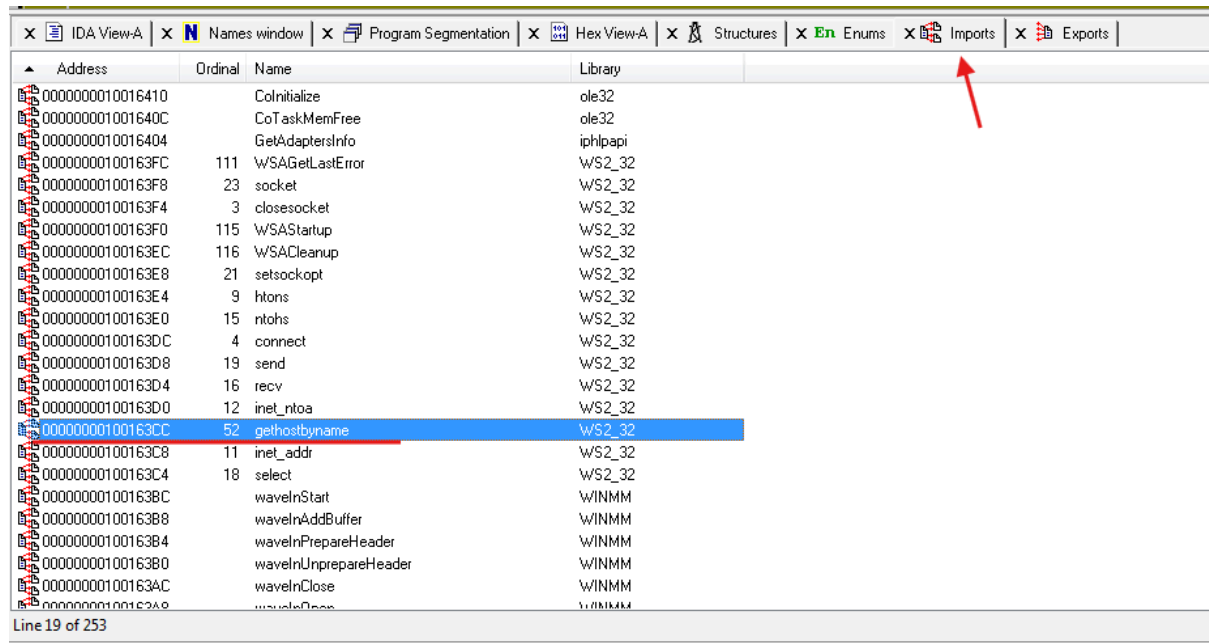


Andiamo a ricercare, tramite il comando “Alt+T” la funzione indicata dalla traccia ed ottenere il codice esadecimale.



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

Spostandoci nella scheda “Imports”, eseguiamo di nuovo la ricerca inserendo “gethostbyname” e trovare così la funzione richiesta dalla traccia con il rispettivo codice esadecimale.



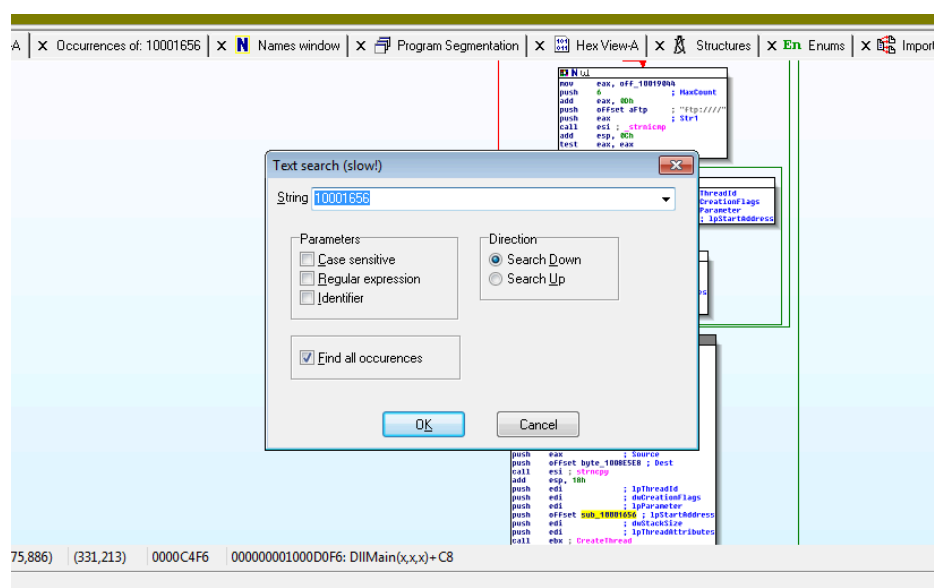
Address	Ordinal	Name	Library
0000000010016410		ColInitialize	ole32
000000001001640C		CoTaskMemFree	ole32
0000000010016404		GetAdaptersInfo	iphlpapi
00000000100163FC	111	WSAGetLastError	WS2_32
00000000100163F8	23	socket	WS2_32
00000000100163F4	3	closesocket	WS2_32
00000000100163F0	115	WSAStartup	WS2_32
00000000100163EC	116	WSACleanup	WS2_32
00000000100163E8	21	setsockopt	WS2_32
00000000100163E4	9	htons	WS2_32
00000000100163E0	15	ntohs	WS2_32
00000000100163DC	4	connect	WS2_32
00000000100163D8	19	send	WS2_32
00000000100163D4	16	recv	WS2_32
00000000100163D0	12	inet_ntoa	WS2_32
00000000100163CC	52	gethostbyname	WS2_32
00000000100163C8	11	inet_addr	WS2_32
00000000100163C4	18	select	WS2_32
00000000100163BC		waveInStart	WINMM
00000000100163B8		waveInAddBuffer	WINMM
00000000100163B4		waveInPrepareHeader	WINMM
00000000100163B0		waveInUnprepareHeader	WINMM
00000000100163AC		waveInClose	WINMM
00000000100163A8		waveInOpen	WINMM

Line 19 of 253

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

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

Come richiesto dalla traccia eseguiamo una ricerca attraverso l'allocazione di memoria indicata.



Aprendo il risultato della ricerca possiamo osservare le variabili ed il parametro.

```
10001656 | X N Names window | X Program Segmentation | X Hex View
; DWORD __stdcall sub_10001656(LPVOID)
sub_10001656 proc near
var_675= byte ptr -675h
var_674= dword ptr -674h
hLibModule= dword ptr -670h
tineout= tineval ptr -66Ch
name= sockaddr ptr -664h
var_654= word ptr -654h
Dst= dword ptr -650h
Parameter= byte ptr -644h
var_640= byte ptr -640h
CommandLine= byte ptr -63Fh
Source= byte ptr -63Dh
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
push esi
push edi
call sub_10001000
test eax, eax
```

00000A56 | 0000000010001656: sub_10001656

5. Inserire altre considerazioni macro livello sul malware (comportamento)

Per ottenere delle considerazioni concrete del comportamento del malware, eseguiamo alcune ricerche all'interno dell'elenco delle funzioni per comprendere meglio il malware.

Cercando la parola chiave “reg” abbiamo come risultato questo gruppo di funzioni che vanno ad eseguire azioni malevole attraverso il registro.

0000000010016058	QueryServiceStatusEx	ADVAPI32
00000000100161EC	ReadFile	KERNEL32
0000000010016098	RealizePalette	GDI32
000000001001637C	RedrawWindow	USER32
0000000010016008	RegCloseKey	ADVAPI32
0000000010016038	RegCreateKeyA	ADVAPI32
000000001001603C	RegDeleteKeyA	ADVAPI32
000000001001601C	RegDeleteValueA	ADVAPI32
0000000010016020	RegEnumKeyA	ADVAPI32
0000000010016030	RegEnumValueA	ADVAPI32
0000000010016024	RegOpenKeyA	ADVAPI32
0000000010016010	RegOpenKeyExA	ADVAPI32
000000001001600C	RegQueryValueExA	ADVAPI32
0000000010016018	RegSetValueExA	ADVAPI32
0000000010016048	RegisterServiceCtrlHandlerA	ADVAPI32
0000000010016394	ReleaseDC	USER32
00000000100161A4	RemoveDirectoryA	KERNEL32

Cercando invece la parola chiave “get” abbiamo un elenco di funzioni maggiore, che vanno ad eseguire azioni di vario genere, ottenendo in base al target, informazioni sensibili differenti.

0000000010016188	FindNextFileA	KERNEL32
00000000100161D8	FreeConsole	KERNEL32
0000000010016210	FreeLibrary	KERNEL32
0000000010016404	GetAdaptersInfo	iphlpapi
0000000010016100	GetComputerNameA	KERNEL32
0000000010016114	GetCurrentDirectoryA	KERNEL32
00000000100160DC	GetCurrentProcess	KERNEL32
0000000010016220	GetCurrentProcessId	KERNEL32
0000000010016118	GetCurrentThreadId	KERNEL32
0000000010016390	GetDC	USER32
0000000010016088	GetDIBits	GDI32
0000000010016360	GetDesktopWindow	USER32
00000000100160B8	GetDeviceCaps	GDI32
00000000100160EC	GetDiskFreeSpaceA	KERNEL32
00000000100160F0	GetDriveTypeA	KERNEL32
000000001001622C	GetExitCodeThread	KERNEL32
00000000100161A0	GetFileAttributesA	KERNEL32
00000000100161AC	GetFileTime	KERNEL32
00000000100160D8	GetLastError	KERNEL32
0000000010016194	GetLocalTime	KERNEL32
00000000100160F4	GetLogicalDrives	KERNEL32
0000000010016378	GetMessageA	USER32
000000001001610C	GetModuleFileNameA	KERNEL32
0000000010016220	GetModuleFileNameExA	PSAPI

Line 145 of 253

000000001001635C	GetProcessWindowStation	USER32
00000000100161F4	GetStartupInfoA	KERNEL32
00000000100160D0	GetStdHandle	KERNEL32
00000000100160A0	GetStockObject	GDI32
0000000010016120	GetSystemDefaultLangID	KERNEL32
00000000100161D0	GetSystemDirectoryA	KERNEL32
0000000010016384	GetSystemMetrics	USER32
00000000100161CC	GetSystemTime	KERNEL32
0000000010016364	GetThreadDesktop	USER32
0000000010016204	GetTickCount	KERNEL32
000000001001639C	GetUserObjectInformationA	USER32
000000001001612C	GetVersion	KERNEL32
00000000100160D4	GetVersionExA	KERNEL32
0000000010016180	GetVolumeInformationA	KERNEL32
00000000100161C8	GetWindowsDirectoryA	KERNEL32
0000000010016140	GlobalAlloc	KERNEL32
0000000010016130	GlobalFree	KERNEL32
000000001001613C	GlobalLock	KERNEL32
00000000100160FC	GlobalMemoryStatus	KERNEL32
0000000010016134	GlobalReAlloc	KERNEL32
0000000010016144	GlobalSize	KERNEL32
0000000010016138	GlobalUnlock	KERNEL32
000000001001630C	ICClos	MSVFW32
0000000010016308	ICCmpress	MSVFW32

Cercando “open” come parola chiave, possiamo vedere che ci sono queste funzioni che hanno come target il desktop.

0000000010016374	PostThreadMessageA	USER32
0000000010016370	PostMessageA	USER32
000000001001611C	OutputDebugStringA	KERNEL32
0000000010016358	OpenWindowStationA	USER32
000000001001607C	OpenServiceA	ADVAPI32
0000000010016070	OpenSCManagerA	ADVAPI32
0000000010016004	OpenProcessToken	ADVAPI32
0000000010016228	OpenProcess	KERNEL32
0000000010016398	OpenInputDesktop	USER32
0000000010016350	OpenDesktopA	USER32
00000000100161E4	MultiByteToWideChar	KERNEL32
0000000010016108	MoveFileExA	KERNEL32
00000000100161A8	MoveFileA	KERNEL32
000000001001614C	Module32Next	KERNEL32
0000000010016150	Module32First	KERNEL32
0000000010016348	MessageBoxA	USER32
0000000010016000	LookupPrivilegeValueA	ADVAPI32
00000000100161C0	LocalFree	KERNEL32
00000000100161C4	LocalAlloc	KERNEL32
00000000100161E0	LoadLibraryA	KERNEL32

In conclusione, dopo una macro analisi delle funzioni possiamo confermare che le intenzioni di questo file malevolo hanno come target molti dati sensibili e molte informazioni che possono compromettere molto l'utilizzo del device e soprattutto la sicurezza e l'incolumità del soggetto attaccato.