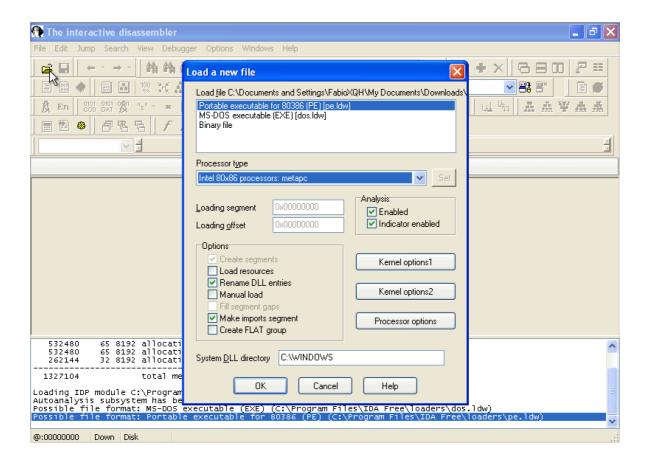
## ANALISI STATICA AVANZATA CON IDA

## 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
- 2. Dalla scheda «imports» individuare la funzione «gethostbyname». Qual è l'indirizzo dell'import?
- 3. Quante sono le variabili locali della funzione alla locazione di memoria 0×10001656?
- 4. Quanti sono, invece, i parametri della funzione sopra?

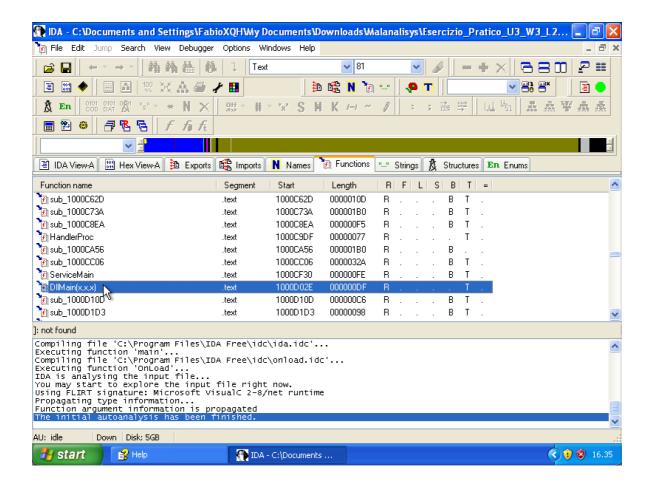
Per prima cosa andremo ad avviare la il programma nella nostra macchina virtuale. Una volta all'interno andremo ad inserire il file che abbiamo intenzione di analizzare, nel nostro caso è: "malware\_U3\_W3\_L2".

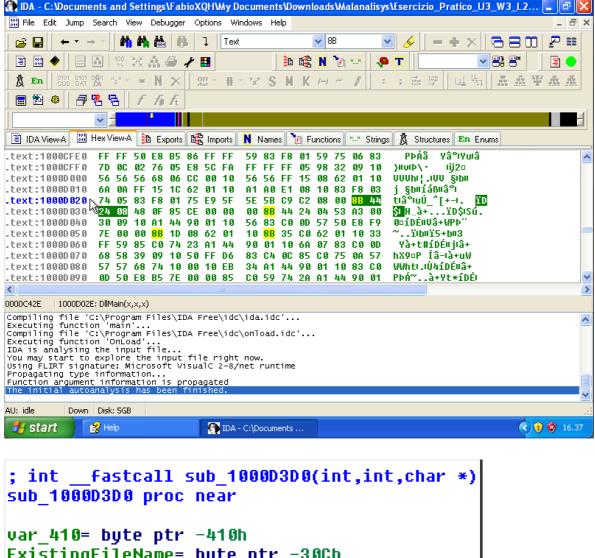


Come possiamo osservare, una volta terminata la scansione il programma ci offre diversi parametri da analizzare.

Indirizzo DLLMain:

Nel parametro funzioni andiamo ad analizzare la funzione DLLMain, la quale ci fornisce il proprio indirizzo 1000Do2E.





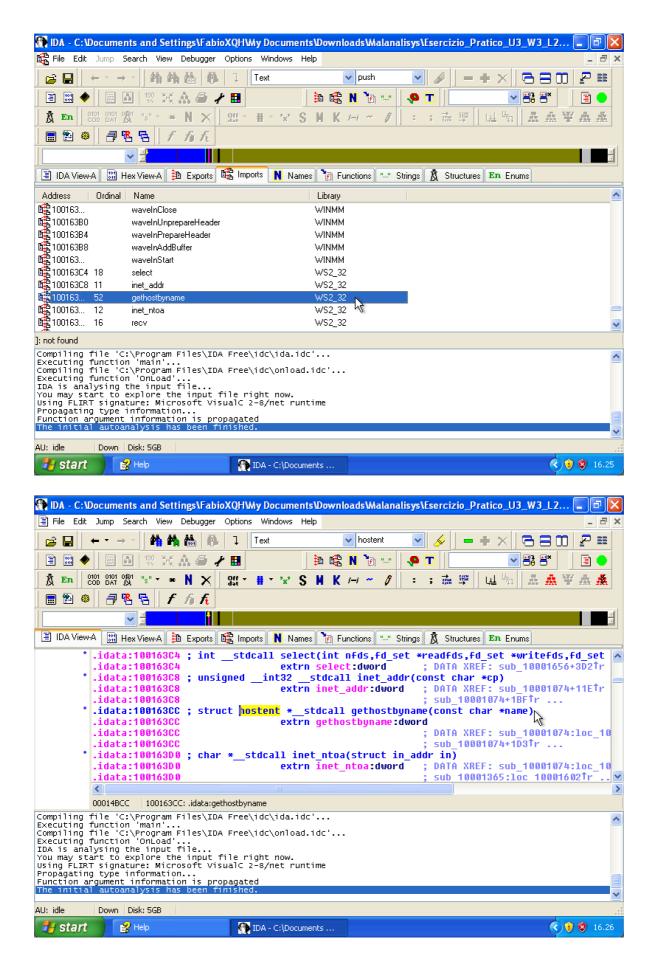
```
; int ___fastcall sub_10000300(int,int,char *)
sub_10000300 proc near

var_410= byte ptr -410h
ExistingFileName= byte ptr -30Ch
NewFileName= byte ptr -208h
var_104= byte ptr -104h
arg_0= dword ptr 8
```

Variabili dell'indirizzo gethostbyname:

Sempre nel parametro imports andiamo ad analizzare la funzione gethostbyname, per cercare di scoprire il suo indirizzo, il quale ci indirizza al indirizzo .idata:100163CC.

Questo indirizzo ci porta alla funzione di "struct hostent", la quale ha il compito di raccogliere informazioni ci l'host della macchina bersaglio.



Variabili all'indirizzo 0x100001656:

Per scoprire quante variabili contenga l'indirizzo 0x10001656, occorre cercarlo nella

sezione jump - jump address e digitare l'indirizzo che si vuole cercare. Nel nostro caso ci è stato mostrato a schermo che le variabili contenuto in questo indirizzo sono 20 in totale ad offset negativo rispetto ad EBP.

```
var 675= byte ptr -675h
var 674= dword ptr -674h
hModule= dword ptr -670h
timeout= timeval ptr -66Ch
name= sockaddr ptr -664h
var 654= word ptr -654h
in= in addr ptr -650h
Parameter= byte ptr -644h
CommandLine= byte ptr -63Fh
Data= byte ptr -638h
var 544= dword ptr -544h
var 50C= dword ptr -50Ch
var 500= dword ptr -500h
var 4FC= dword ptr -4FCh
readfds= fd set ptr -4BCh
phkResult= HKEY__ ptr -3B8h
var 380= dword ptr -380h
var 1A4= dword ptr -1A4h
var 194= dword ptr -194h
WSAData= WSAData ptr -190h
arq 0= dword ptr
```

Mentre per quanto riguarda i parametri ne abbiamo riscontrato solo uno in positivo rispetto ad EBPl, ed è il seguente:

Per quanto riguarda il comportamento di questo malware, si può ipotizzare dalle immagini che esso raccolga le chiavi di registro del sistema (call ds:RegOpenKeyExa), per poi andarlo a modificare attraverso (call:RegSetValueExa).

```
cdecl sub 1000564E(LPCSTR lpValueName, BYTE Data)
sub 1000564E proc near
hKey= dword ptr -4
lpValueName= dword ptr 8
Data= byte ptr OCh
push
        ebp
mnv
        ebp, esp
push
        ecx
lea
        eax, [ebp+hKey]
nush
        esi
push
                         ; phkResult
        eax
        esi, esi
xor
        0F 0 03Fh
                         ; samDesired
push
push
        esi
                          ulOptions
        offset aSoftwareMicros ; "SOFTWARE\\Microsoft\\Windows\\CurrentVersi"...
push
        80000002h
push
                         ; hKey
call
        ds:ReqOpenKeyExA
test
        eax, eax
        short loc_1000568F
jnz
```

```
🛗 N 👊
        eax, [ebp+Data]
lea
push
        4
                           cbData
push
                           lpData
        eax
push
                            dwType
push
        esi
                            Reserved
        [ebp+lpValueName] ; lpValueName
push
        [ebp+hKey]
push
                            hKey
call
        ds:ReqSetValueExA
test
        eax, eax
        short loc 1000568F
jnz
```

A tal proposito, per avere maggiori dettagli siamo andati a calcolare gli hash del file del malware e caricarli sul sito di virus total il quale ci ha fornito l'identità di questo malware, ovvero di una backdoor.

```
"--" xdoors_d:10095B44
                  00000118
                                  Hi.Master [%d/%d/%d %d.%d.%d]\\r\w\elCome Back...Are You Enjoying Today?\\r\\r\Machine UpTime [%-2d Days %-2d Hours %-2d Minutes %-2d Seconds]\\r\Machine I
  xdoors_d:10093740
                  000000AD
                                 xdoors_d:10093AFC
                  00000055
                                 "--" xdoors_d:10093F94
                  00000054
  xdoors_d:10093F40
                  00000051
"--" xdoors d:10095820
                  00000050
                                 MayBe Inject Mode. You Only Need Close Master Process '%s' To Uninatil This Mode
                  0000004F
"--" .data:100191E4
                  0000004F
                                 [This is BIP]
"--" .data:10019054
                  0000004F
"--" xdoors_d:10093ED4
                  0000004D
                                 \r\n\r\n[Machine UpTime:1%-.2d Days %-.2d Hours %-.2d Minutes %-.2d Seconds\r\n\r\n
```

https://www.virustotal.com/gui/file/eb1079bdd96bc9cc19c38b76342113a09666aad47 518ff1a7536eebff8aadb4a/detection