Analisi statica avanzata

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)

L'analisi statica avanzata si concentra sull'esame approfondito del codice sorgente o dei file eseguibili senza eseguirli effettivamente.

1)

```
.text:1000D02E
.text:1000D02E ; ||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||||||||||||||
.text:1000D02E
.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
.text:1000D02E                 mov     eax, [esp+fdwReason]
.text:1000D032                 dec     eax
.text:1000D033                 jnz     loc_1000D107
```

hinstDLL = dword ptr 4: Sta dichiarando una variabile chiamata hinstDLL e le sta assegnando il valore a 4 byte (dword) situato all'indirizzo di memoria 4.

fdwReason = dword ptr 8: Sta dichiarando una variabile chiamata fdwReason e le sta assegnando il valore a 4 byte (dword) situato all'indirizzo di memoria 8.

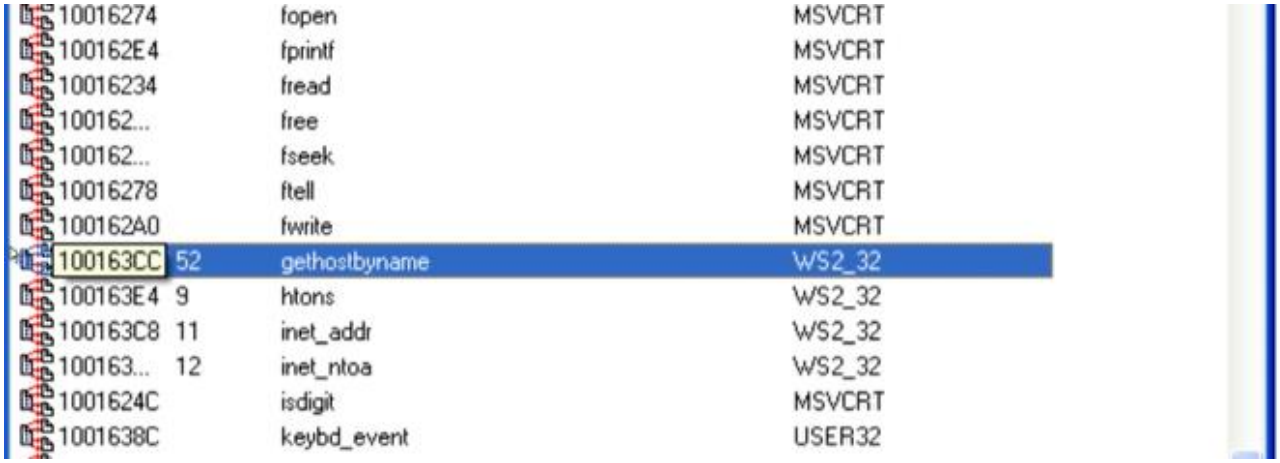
lpvReserved = dword ptr 0Ch: Sta dichiarando una variabile chiamata lpvReserved e le sta assegnando il valore a 4 byte (dword) situato all'indirizzo di memoria 0Ch (12 in decimale).

mov eax, [esp+1000b]: Carica il valore situato all'indirizzo di memoria [esp+1000b] nel registro EAX.

dec eax: Decrementa il valore nel registro EAX di 1.

jnz loc_1000D107: Salta all'indirizzo loc_1000D107 se il flag di zero (ZF) non è impostato. In altre parole, se il risultato del decremento non è zero, salta.

2) GETHOSTBYNAME: La chiamata "call" restituisce il nome e l'indirizzo IP di un host il cui nome del dispositivo. Un dato host TCP/IP può avere diversi nomi e diversi indirizzi Internet dell'host.



Il malware potrebbe connettersi a un indirizzo remoto

3) 4)

Come vediamo dall'immagine sono 20 variabili e un parametro
(il solo e unico parametro è)

```
; DWORD __stdcall sub_10001656(LPVOID)
sub_10001656 proc near

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_3B0= dword ptr -3B0h
var_1A4= dword ptr -1A4h
var_194= dword ptr -194h
WSAData= WSAData ptr -190h
arg_0= dword ptr  4
```

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

Con offset si intende la differenza rispetto ad un valore di riferimento

5) Per scoprire di che tipo di malware si trattasse abbiamo preso il codice hash fornitoci da CFF

MD5	1A9FD80174AAFECD9A52FD908CB82637
SHA-1	FBE285B8B7FE710724EA35D15948969A709ED33B

Infine l'abbiamo incollato su virus total, da li abbiamo riconosciuto che si trattasse di una **backdoor**

```
xdoors_d:10093D50                db '(1) Enter Current Directory ',27h,'%s',27h,0
xdoors_d:10093D73                align 4
xdoors_d:10093D74 ; char aBackdoorServer[]
xdoors_d:10093D74 aBackdoorServer db 0Dh,0Ah             ; DATA XREF: sub_100042DB+B5↑o
xdoors_d:10093D74                db 0Dh,0Ah
xdoors_d:10093D74                db '*****************************',0Dh,0Ah
xdoors_d:10093D74                db '[BackDoor Server Update Setup]',0Dh,0Ah
xdoors_d:10093D74                db '*****************************',0Dh,0Ah
xdoors_d:10093D74                db 0Dh,0Ah,0
xdoors_d:10093DDB                align 4
xdoors_d:10093DDC ; char aWarn[]
```

Infine riesaminando le linee di codice si può notare che cè una connessione backdoor a un server