

Oggi andremo ad analizzare il malware dato dall'esercizio con ida pro, un noto dissassembler. Una volta caricato su ida, cerchiamo la funzione dll main del codice:

```

-A
.text:1000D02E ; SUBROUTINE
.text:1000D02E
.text:1000D02E
.text:1000D02E ; BOOL __stdcall DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPOVOID lpvRese
.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

```

0000C42E 1000D02E: DllMain(x,x,x)

In questo modo ne ricaviamo anche l'indirizzo di memoria :

```

F DllMain(x,x,x) 1000D02E

```

Una volta trovata, da imports individuiamo la funzione "gethostbyname:

```

.idata:100163CC ; struct hostent *__stdcall gethostbyname(const char *name)
.idata:100163CC      extrn gethostbyname:dword
.idata:100163CC                                     ; DATA XREF: sub_10001074:loc_100
.idata:100163CC                                     ; sub_10001074+1D3↑r ...

```

E ne ricaviamo l'indirizzo: I gethostbyname 100163CC

Dopo aver cercato l'indirizzo di memoria 0x10001656 abbiamo trovato le variabili locali della funzione alla suddetto indirizzo:

<pre> 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 </pre>	<pre> 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 </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Queste sono tutte variabili locali della funzione in quanto, come studiato nella lezione della mattina, hanno un offset negativo rispetto ad ebx.

Invece l'unico parametro all'interno della funzione che stiamo studiando è il seguente:

```
|arg_0= dword ptr 4
```

In quanto è l'unico ad avere offset positivo.

Infine abbiamo appurato che il malware in questione è una backdoor, in quanto, dopo aver effettuato un'analisi statica basica ed aver estrapolato il suo hash ed averlo cercato su virus total, il risultato è il seguente:

DETECTION			DETAILS	RELATIONS	BEHAVIOR	COMMUNITY
Popular threat label			Threat categories			trojan
Security vendors' analysis						
AhnLab-V3	Backdoor/Win32.Agent.R9408	Alibaba				
ALYac	Backdoor.XIW	Antiy-AVL				
Arcabit	Backdoor.XIW	Avast				