

Aorile 2024

CYBER SECURITY SPECIALIST

S11L2

Jacopo Trovato

Analisi statica avanzata con IDA

Attraverso il tool "IDA" è possibile analizzare dei malware. In questo caso verrà analizzato il file "Malware_U3_W3_L2". Una volta inserito il malware nel tool, cercare la funzione "Dllmain".

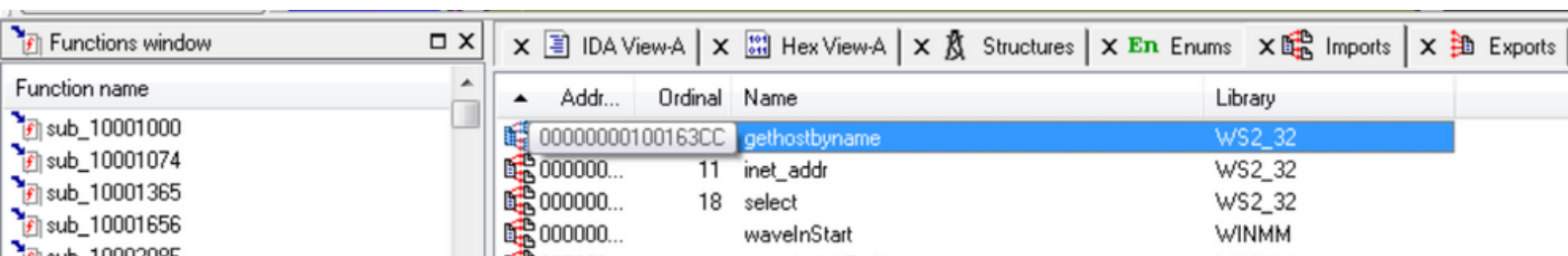
```
; BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
DllMain@12 proc near

hinstDLL= dword ptr 4
fdwReason= dword ptr 8
lpvReserved= dword ptr 0Ch

mov     eax, [esp+fdwReason]
dec     eax
jnz     loc_1000D107
```

Funzione DllMain
nel file
Malware_U3_W3_L2

Andare nella sezione "Imports" cercare la funzione "gethostbyname" e trovare il suo indirizzo e il suo scopo, ovvero individuare l'"indirizzo IP" di un host.



Function name	Addr...	Ordinal	Name	Library
sub_10001000				
sub_10001074				
sub_10001365				
sub_10001656				
sub_1000208F				
	00000000100163CC		gethostbyname	WS2_32
	00000000...	11	inet_addr	WS2_32
	00000000...	18	select	WS2_32
	00000000...		waveInStart	WINMM

Indirizzo della
funzione gethostbyname

Nella sezione "Function Window" andare nella locazione di memoria "0x10001656", e controllare il numero di variabili, in questo caso 11. Successivamente andare in quella superiore "0x10001365" e controllare in numero di parametri, in questo caso 2:

- File = rappresenta un file che deve essere aperto, letto, scritto o chiuso attraverso operazioni di input/output (I/O) di basso livello.
- Dst = "destination" (destinazione) indica il luogo dove devono essere memorizzati i risultati di un'operazione.

```

var_675= byte ptr -675h
var_674= dword ptr -674h
hLibModule= dword ptr -670h
timeout= ttimeval 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 -48Ch
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

```

Variabili della locazione
di memoria
0x10001656

```

; DWORD __stdcall sub_10001365(LPVOID)
sub_10001365 proc near

```

```

File= FILE ptr -54h
var_30= word ptr -30h
in= in_addr ptr -2Ch
Dst= byte ptr -20h
var_1F= byte ptr -1Fh

```

Parametri della
locazioni di memoria
0x10001365

```

sub     esp, 54h
push    ebx
push    ebp
push    esi
push    edi
call    sub_10001000
test    eax, eax
jz      short loc_10001381

```

Analizzando il malware è probabile che questo si tratti di una **"backdoor"**, tipo di software dannoso progettato per consentire a un attaccante di accedere segretamente e controllare un computer o una rete senza l'autorizzazione dell'utente. Si deduce perchè si trovano funzioni e variabili inerenti ad una **"backdoorserver"**

```

push    edi                ; nBufferLength
call    ds:GetCurrentDirectoryA
mov     esi, ds:sprintf
lea     eax, [ebp+buf]
push    offset aBackdoorServer ; "\\r\n\r\n*****\r\n[Ba"...
push    eax                ; Dest
call    esi ; sprintf
mov     ebx, [ebp+s]
lea     eax, [ebp+buf]
push    eax                ; buf
push    ebx                ; s
call    sub_100038BB
add     esp, 10h
lea     eax, [ebp+PathName]
push    eax                ; lpPathName
call    ds:SetCurrentDirectoryA
test    eax, eax
jz      loc_100046E1

```

Backdoorserver a cui fanno
riferimento alcune variabili e funzioni