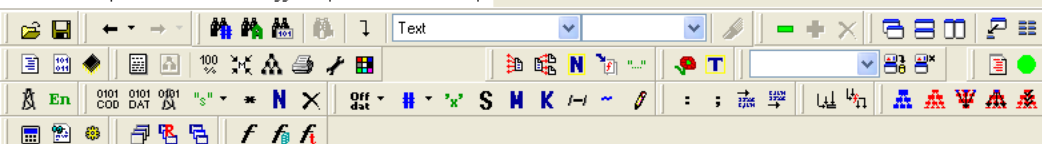


Troviamo l'indirizzo di memoria della funzione DDLMain in esadecimale (1000D02E)

IDA - C:\Documents and Settings\Administrator\Desktop\Esercizio_Pratico_U3_W3_L2\Malware_U3_W3_L2.dll

File Edit Jump Search View Debugger Options Windows Help



IDA View-A Hex View-A Exports Imports Names Functions Strings Structures Enums

IDA View-A

```
; B00L __stdcall DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPUVOID 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
```

```
mov     eax, [esp+hinstDLL]
push    ebx
mov     ds:hModule, eax
mov     eax, off_10019044
push    esi
add     eax, 0Dh
push    edi
push    eax
call    j_strlen
mov     ebx, ds:CreateThread
mov     esi, ds:_strnicmp
xor     edi, edi
pop     ecx
test    eax, eax
jz      short loc_1000D089
```

100.00% (-463,-15) (1306,0) 0000C433 1000D033: DllMain(x,x,x)+5

Executing function 'main'...
Compiling file 'C:\Program Files\IDA Free\idc\onload.idc'...
Executing function 'OnLoad'...
IDA is analysing the input file...
You may start to explore the input file right now.
Using FLIRT signature: Microsoft VisualC 2-8/net runtime
Propagating type information...
Function argument information is propagated.
The initial autoanalysis has been finished.
Retrieving information from the database... ok

AU: idle Down Disk: 53GB

start Help IDA - C:\Documents ...

N Names window

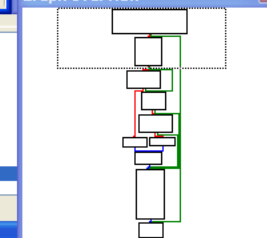
Name	Address	P
CreateServiceA	10016050	
CreateThread	10016208	
CreateToolhelp32Snapshot	100160E8	
CriticalSection	1001D878	
DeleteDC	10016084	
DeleteFileA	10016160	
DeleteObject	100160C0	
DeleteService	1001606C	
DllEntryPoint	1001516D	P
DllMain(x,x,x)	1000D02E	
DrawIextA	10016380	
DuplicateTokenEx	1001602C	
EnterCriticalSection	10016178	

Line 41 of 761

Strings window

Address	Length	T...	String
["...".rdata:1...	00000005	C	vids
["...".rdata:1...	0000000C	C	SHELL32.dll
["...".rdata:1...	00000009	C	DeleteDC
["...".rdata:1...	0000000D	C	DeleteObject
["...".rdata:1...	0000000A	C	GetDIBits
["...".rdata:1...	0000000F	C	RealizePalette
["...".rdata:1...	0000000E	C	SelectPalette
["...".rdata:1...	0000000F	C	GetStockObject
["...".rdata:1...	0000000B	C	GetObjectA
["...".rdata:1...	00000007	C	BitBlt
["...".rdata:1...	0000000D	C	SelectObject
["...".rdata:1...	00000017	C	CreateCompatibleRgn

Graph overview



Individuiamo la funzione "gethostbyname" tramite la sezione imports (indirizzo 100163CC).

IDA - C:\Documents and Settings\Administrator\Desktop\Esercizio_Pratico_U3_W3_1.2\malware_U3_W3_1.2.dll

File Edit Jump Search View Debugger Options Windows Help

IDA View-A

```
.rdata:100163C4 ; int __stdcall select(int nfds,fd_set *readfds,fd_set *writefds,fd_set *exceptfds,const struct timeval *timeout)
.rdata:100163C4 select      dd ? ; DATA XREF: sub_10001656+3D21r
.rdata:100163C8 ; unsigned __int32 __stdcall inet_addr(const char *cp)
.rdata:100163C8 inet_addr   dd ? ; DATA XREF: sub_10001074+11E1r
.rdata:100163CC ; struct hostent * __stdcall gethostbyname(const char *name)
.rdata:100163CC gethostbyname dd ? ; DATA XREF: sub_10001074:loc_100011AF1r
.rdata:100163D0 ; char * __stdcall inet_ntoa(struct in_addr in)
.rdata:100163D0 inet_ntoa   dd ? ; DATA XREF: sub_10001074:loc_100013111r
.rdata:100163D4 ; int __stdcall recv(SOCKET s,char *buf,int len,int flags)
.rdata:100163D4 recv       dd ? ; DATA XREF: sub_10001656+2D51r
.rdata:100163D8 ; int __stdcall send(SOCKET s,const char *buf,int len,int flags)
.rdata:100163D8 send       dd ? ; DATA XREF: sub_10001656+2901r
.rdata:100163DC ; int __stdcall connect(SOCKET s,const struct sockaddr *name,int namelen)
.rdata:100163DC connect    dd ? ; DATA XREF: sub_10001656+2511r
.rdata:100163E0 ; u_short __stdcall ntohs(u_short netshort)
.rdata:100163E0 ntohs      dd ? ; DATA XREF: sub_10001656+2141r
.rdata:100163E4 ; u_short __stdcall htons(u_short hostshort)
.rdata:100163E4 htons      dd ? ; DATA XREF: sub_1000208F+3821r
.rdata:100163E8 ; int __stdcall setsockopt(SOCKET s,int level,int optname,const char *optval,int optlen)
.rdata:100163E8 setsockopt dd ? ; DATA XREF: sub_1000208F+42F1r
.rdata:100163EC ; int WSACleanup(void)
.rdata:100163EC WSACleanup dd ? ; DATA XREF: sub_1000208F:loc_10002CB41r
.rdata:100163F0 ; int __stdcall WSAStartup(WORD wVersionRequested,LPWSAData lpWSAData)
.rdata:100163F0 WSAStartup dd ? ; DATA XREF: sub_10001656+4E1r
```

Imports

Address	Ordinal	Name	Library
10016288		_strrev	MSVCRT
100162E8		_strtime	MSVCRT
10016258		_strupr	MSVCRT
100162E0		_vsprintf	MSVCRT
10016268		abs	MSVCRT
100162B4		atoi	MSVCRT
100163F4	3	closesocket	WS2_32
100163DC	4	connect	WS2_32
100162A4		fclose	MSVCRT
10016274		fopen	MSVCRT
100162E4		fprintf	MSVCRT
10016234		fread	MSVCRT
100162DC		free	MSVCRT
100162D8		fseek	MSVCRT
10016278		ftell	MSVCRT
100162A0		fwrite	MSVCRT
100163CC	52	gethostbyname	WS2_32
100163E4	9	htons	WS2_32
100163C8	11	inet_addr	WS2_32
100163D0	12	inet_ntoa	WS2_32
1001624C		isdigit	MSVCRT
1001638C		keybd_event	USER32
10016264		malloc	MSVCRT
100162AC		memcmp	MSVCRT
100162C8		memcpy	MSVCRT
100162D4		memset	MSVCRT
10016388		mouse_event	USER32
100163E0	15	ntohs	WS2_32
10016268		printf	MSVCRT

Names window

Length	T...	String
00000012	C	GetExitCodeThread
00000013	C	GetFileAttributesA
00000000C	C	GetFileTime
00000000D	C	GetLastError
000000011	C	GetLastInputInfo
00000000D	C	GetLocalTime
000000011	C	GetLogicalDrives
00000000C	C	GetMessageA
000000021	C	GetModuleFileName() get dll path
000000013	C	GetModuleFileNameA
000000015	C	GetModuleFileNameExA
nnnnnn11	C	GetModuleHandleA

Executing function 'OnLoad'...

IDA is analysing the input file...

You may start to explore the input file right now.

Using FLIRT signature: Microsoft VisualC 2-8/net runtime

Propagating type information...

Function argument information is propagated

The initial autoanalysis has been finished.

Retrieving information from the database... ok

Retrieving information from the database... ok

Retrieving information from the database... ok

AU: idle Down Disk: 53GB

start Help IDA - C:\Documents ...

2:02 PM

Per ultimo, possiamo notare le variabili locali ed i parametri all'indirizzo di memoria 0x10001656.

Le variabili locali sono in totale 20, riconoscibili dall'offset negativo (esempio; -675h/-190h).

L'offset è la differenza, positiva o negativa, dal valore di riferimento nella variabile.

I parametri, al contrario, hanno un offset positivo. Ce n'è uno soltanto ed è "arg_0" con offset di 4.

Possiamo anche dedurre che sia una backdoor poiché all'interno del codice si trovano alcune variabili inerenti ad un ipotetico server di backdoor.

```
.text:10001656  
.text:10001656 var_675      = byte ptr -675h  
.text:10001656 var_674      = dword ptr -674h  
.text:10001656 hModule      = dword ptr -670h  
.text:10001656 timeout      = timeval ptr -66Ch  
.text:10001656 name          = sockaddr ptr -664h  
.text:10001656 var_654      = word ptr -654h  
.text:10001656 in            = in_addr ptr -650h  
.text:10001656 Parameter      = byte ptr -644h  
.text:10001656 CommandLine    = byte ptr -63Fh  
.text:10001656 Data          = byte ptr -638h  
.text:10001656 var_544      = dword ptr -544h  
.text:10001656 var_50C      = dword ptr -50Ch  
.text:10001656 var_500      = dword ptr -500h  
.text:10001656 var_4FC      = dword ptr -4FCh  
.text:10001656 readfds       = fd_set ptr -4BCh  
.text:10001656 phkResult     = HKEY__ ptr -3B8h  
.text:10001656 var_3B0      = dword ptr -3B0h  
.text:10001656 var_1A4      = dword ptr -1A4h  
.text:10001656 var_194      = dword ptr -194h  
.text:10001656 WSADATA       = WSADATA ptr -190h  
.text:10001656 arg_0         = dword ptr 4  
.text:10001656
```

```
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
```