

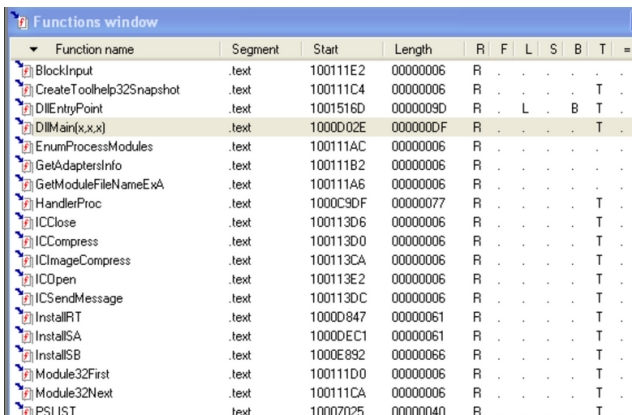
Esercizio S11L2

Traccia:

Lo scopo dell'esercizio di oggi è di acquisire esperienza con IDA, un tool fondamentale per l'analisi statica.

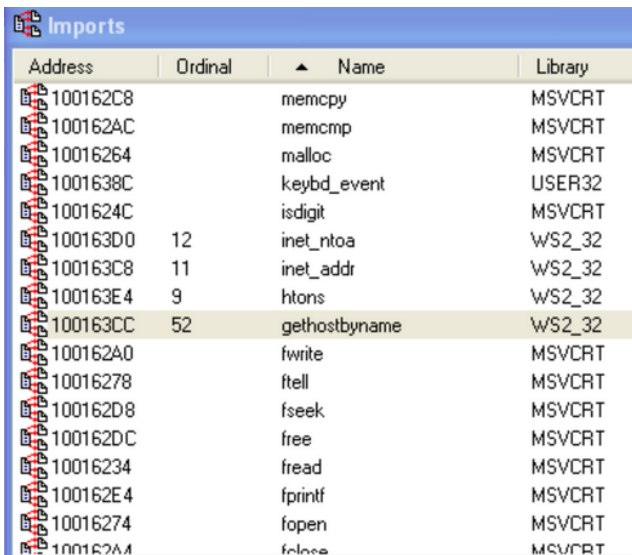
A tal proposito, con riferimento al malware chiamato «Malware_U3_W3_L2» presente all'interno della cartella «Esercizio_Pratico_U3_W3_L2» sul Desktop della macchina virtuale dedicata all'analisi dei malware, rispondere ai seguenti quesiti, utilizzando IDA Pro.

1. Individuare l'indirizzo della funzione **DLLMain** (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 a macro livello sul malware (comportamento)



Function name	Segment	Start	Length	R	F	L	S	B	T
BlockInput	.text	100111E2	00000006	R
CreateToolhelp32Snapshot	.text	100111C4	00000006	R	T
DllEntryPoint	.text	1001516D	0000009D	R	.	L	.	B	T
DLLMain(x,x,x)	.text	1000D02E	0000000F	R	T
EnumProcessModules	.text	100111AC	00000006	R
GetAdapterInfo	.text	100111B2	00000006	R
GetModuleFileNameExA	.text	100111A6	00000006	R
HandlerProc	.text	1000C9DF	00000077	R	T
ICClose	.text	100113D6	00000006	R	T
ICCompress	.text	100113D0	00000006	R	T
ICImageCompress	.text	100113CA	00000006	R	T
ICOpen	.text	100113E2	00000006	R	T
ICSendMessage	.text	100113DC	00000006	R	T
InstallIRT	.text	1000D847	00000061	R	T
InstallISA	.text	1000DEC1	00000061	R	T
InstallSB	.text	1000E892	00000066	R	T
Module32First	.text	100111D0	00000006	R	T
Module32Next	.text	100111CA	00000006	R	T
PSLIST	.text	10007025	00000040	R	T

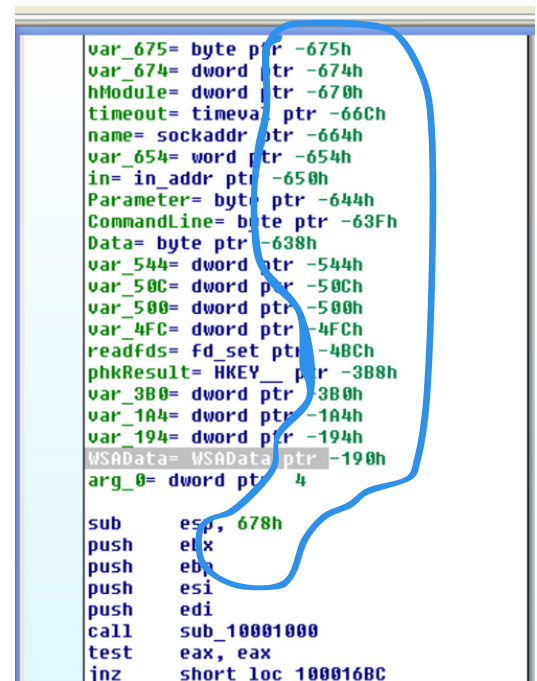
1) L'indirizzo della funzione **DLLMain** corrisponde alla colonna start



Address	Ordinal	Name	Library
100162C8		memcpy	MSVCRT
100162AC		memcmp	MSVCRT
10016264		malloc	MSVCRT
1001638C		keybd_event	USER32
1001624C		isdigit	MSVCRT
100163D0	12	inet_ntoa	WS2_32
100163C8	11	inet_addr	WS2_32
100163E4	9	htons	WS2_32
100163CC	52	gethostbyname	WS2_32
100162A0		fwrite	MSVCRT
10016278		ftell	MSVCRT
100162D8		fseek	MSVCRT
100162DC		free	MSVCRT
10016234		fread	MSVCRT
100162E4		fprintf	MSVCRT
10016274		fopen	MSVCRT
100162A4		fclose	MSVCRT

2) L'indirizzo della funzione **gethostbyname** corrisponde alla colonna address e questa funzione permette di ottenere informazioni su un host specificato dal suo nome.

3) Con la traduzione di IDA Pro del linguaggio macchina in linguaggio assembly ci ha permesso di individuare i valori locali e i parametri che corrispondono tutti i valori con il segno "-" sono variabili locali e senza il segno "-" sono parametri



```
var_675= byte ptr -675h
var_674= dword ptr -674h
hModule= dword ptr -670h
timeout= timevar 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 -48Ch
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

sub     esp, 678h
push    eax
push    ebx
push    esi
push    edi
call    sub_10001000
test    eax, eax
jnz     short loc_100016BC
```

59 / 71

59 security vendors and no sandboxes flagged this file as malicious

Reanalyze Similar More

eb1079dd96bc9cc19c38b76342113a0966aad47518f1a7536eebf8baadb4a

X-doorc

Size: 130.94 KB

Last Analysis Date: 21 hours ago

peid corrupt armadillo overlay

Community Score

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan:ldcafr!06cc0d321 Threat categories: trojan Family labels: ldcafr!06cc0d321

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Backdoor:Win32.Agent.R9408	Alibaba	Backdoor:Win32!ldcafr!9f3a5556
ALYac	Backdoor:XW	Antiy-AVL	Trojan(Backdoor)/Win32.Agent
Arcabit	Backdoor:XW	Avast	Win32-Agent-OLH [Trj]
AVG	Win32-Agent-OLH [Trj]	Avira (no cloud)	BDS!Agent.twe.134160

CFF Explorer VIII - [Malware_U3_W3_L2.dll]

File Settings

Malware_U3_W3_L2.dll

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
000163A6	N/A	000152D0	000152D4	000152D8	000152DC	000152E0
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
ADVAPI32.dll	32	0001685C	00000000	00000000	000178A6	00016000

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00017944	00017944	014D	LookupPrivilegeValueA
0001795C	0001795C	01AA	OpenProcessToken
00017970	00017970	01C9	RegCloseKey
0001797E	0001797E	01EC	RegQueryValueExA
00017992	00017992	01E2	RegOpenKeyExA
000179A2	000179A2	005F	CreateProcessAsUserA
000179BA	000179BA	01F9	RegSetValueExA
000179CC	000179CC	01D2	RegDeleteValueA
000179DE	000179DE	01D5	RegEnumKeyA
000179EC	000179EC	01E1	RegOpenKeyA
000179FA	000179FA	023B	SetTokenInformation
00017A10	00017A10	00B4	DuplicateTokenEx
00017A24	00017A24	01D9	RegEnumValueA
0001792C	0001792C	001C	AdjustTokenPrivileges
00017A34	00017A34	01CC	RegCreateKeyA
00017A44	00017A44	01D0	RegDeleteKeyA
00017A54	00017A54	003E	CloseServiceHandle
00017A6A	00017A6A	01BC	QueryServiceConfigA
00017888	00017888	0201	RegisterServiceCtrlHandlerA
00017874	00017874	0239	SetServiceStatus
00017862	00017862	0064	CreateServiceA
0001784A	0001784A	0034	ChangeServiceConfig2A
00017832	00017832	01C2	QueryServiceStatusEx
0001781A	0001781A	0036	ChangeServiceConfigA
0001780A	0001780A	023E	StartServiceA
00017AF4	00017AF4	01C1	QueryServiceStatus
00017AE2	00017AE2	0042	ControlService
00017AD2	00017AD2	00AF	DeleteService
00017AC0	00017AC0	01AB	OpenSCManagerA
00017AA8	00017AA8	00D2	EnumServicesStatusExA
00017A80	00017A80	01BA	QueryServiceConfig2A
00017A98	00017A98	01AD	OpenServiceA

IDA View-A

```

* xdoors_d:10093D29 align 4
* xdoors_d:10093D2C ; char a_ubak[]
xdoors_d:10093D2C a_ubak db '.ubak',0 ; DATA XREF: sub_100042D8+191f0
* xdoors_d:10093D32 align 4
* xdoors_d:10093D34 ; char a2GetDllFileName[]
xdoors_d:10093D34 a2GetDllFileName db 00h,0Ah ; DATA XREF: sub_100042D8+163f0
xdoors_d:10093D34 db '(2) Get DLL FileName ',27h,'%s',27h,0
* xdoors_d:10093D50 ; char a1EnterCurrentD[]
xdoors_d:10093D50 a1EnterCurrentD db 00h,0Ah ; DATA XREF: sub_100042D8+F2f0
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 00h,0Ah ; DATA XREF: sub_100042D8+B5f0
xdoors_d:10093D74 db 00h,0Ah
xdoors_d:10093D74 db '*****',00h,0Ah
xdoors_d:10093D74 db '[BackDoor Server Update Setup]',00h,0Ah
xdoors_d:10093D74 db '*****',00h,0Ah
xdoors_d:10093D74 db 00h,0Ah,0
* xdoors_d:10093DDB align 4
* xdoors_d:10093DDC ; char aWarn[]
xdoors_d:10093DDC aWarn db '-warn',0 ; DATA XREF: sub_10004738+198f0

```

4) Da quello che si può vedere dai risultati dell'analisi si può supporre che il malware in questione sia una backdoor