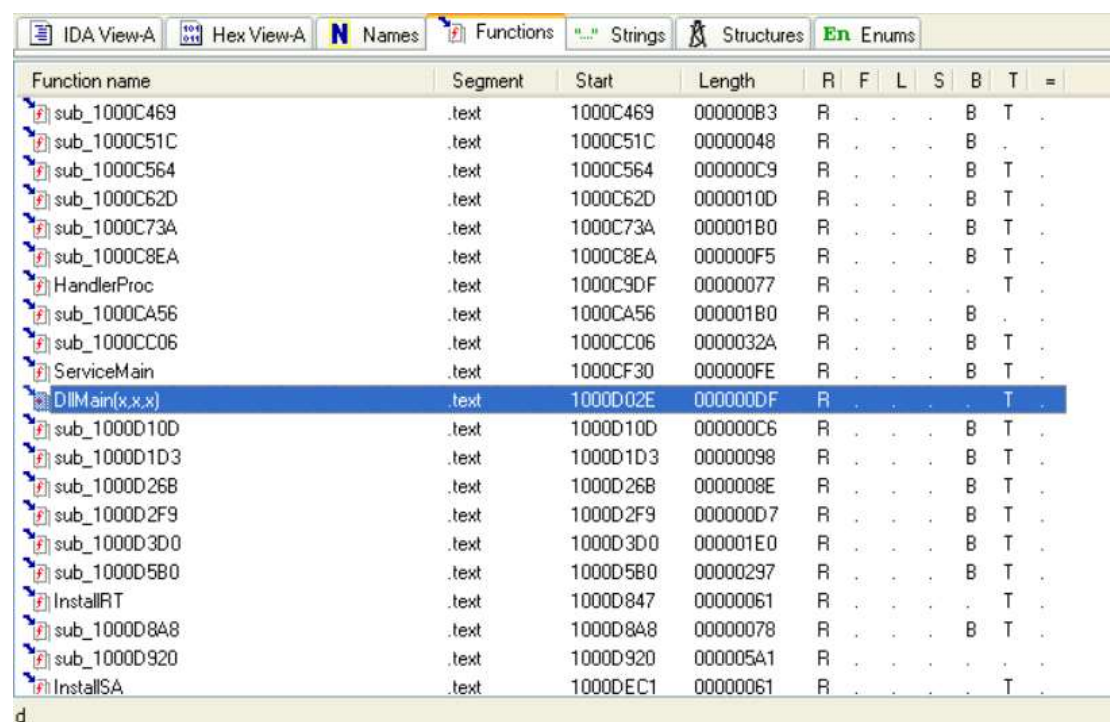


Pratica S11/L2 - Analisi statica avanzata con IDA

L'esercizio di oggi consiste nel rispondere ad alcuni quesiti dopo la scansione di un malware con IDA pro, i quesiti sono i seguenti:

1. Individuare l'indirizzo della funzione DLLMain
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)

1. Individuare l'indirizzo della funzione DLLMain



Function name	Segment	Start	Length	R	F	L	S	B	T	=
sub_1000C469	.text	1000C469	00000083	R	.	.	.	B	T	.
sub_1000C51C	.text	1000C51C	00000048	R	.	.	.	B	.	.
sub_1000C564	.text	1000C564	000000C9	R	.	.	.	B	T	.
sub_1000C62D	.text	1000C62D	0000010D	R	.	.	.	B	T	.
sub_1000C73A	.text	1000C73A	00000180	R	.	.	.	B	T	.
sub_1000C8EA	.text	1000C8EA	000000F5	R	.	.	.	B	T	.
HandlerProc	.text	1000C9DF	00000077	R	T	.
sub_1000CA56	.text	1000CA56	00000180	R	.	.	.	B	.	.
sub_1000CC06	.text	1000CC06	0000032A	R	.	.	.	B	T	.
ServiceMain	.text	1000CF30	000000FE	R	.	.	.	B	T	.
DllMain(x,x,x)	.text	1000D02E	000000DF	R	T	.
sub_1000D10D	.text	1000D10D	000000C6	R	.	.	.	B	T	.
sub_1000D1D3	.text	1000D1D3	00000098	R	.	.	.	B	T	.
sub_1000D26B	.text	1000D26B	0000008E	R	.	.	.	B	T	.
sub_1000D2F9	.text	1000D2F9	000000D7	R	.	.	.	B	T	.
sub_1000D3D0	.text	1000D3D0	000001E0	R	.	.	.	B	T	.
sub_1000D5B0	.text	1000D5B0	00000297	R	.	.	.	B	T	.
InstallRT	.text	1000D847	00000061	R	T	.
sub_1000D8A8	.text	1000D8A8	00000078	R	.	.	.	B	T	.
sub_1000D920	.text	1000D920	000005A1	R
InstallSA	.text	1000DEC1	00000061	R	T	.

```

.text:1000D02B ServiceMain    retn     8
.text:1000D02B ServiceMain    endp
.text:1000D02E ; SUBROUTINE
.text:1000D02E
.text:1000D02E ; BOOL __stdcall DLLMain(HINSTANCE hinstDLL,DWORD fdwReason,LPOVOID lpvReserved)
.text:1000D02E _DllMain@12    proc near    ; CODE XREF: DllEntryPoint+4B1p
.text:1000D02E                                     ; DATA XREF: sub_100110FF+2D10
.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:1000D032 jnz     loc_1000D107
.text:1000D039 mov     eax, [esp+hinstDLL]
.text:1000D03D push    ebx
.text:1000D03E mov     ds:hModule, eax

```

L'indirizzo della funzione DLLMain è 1000D02E

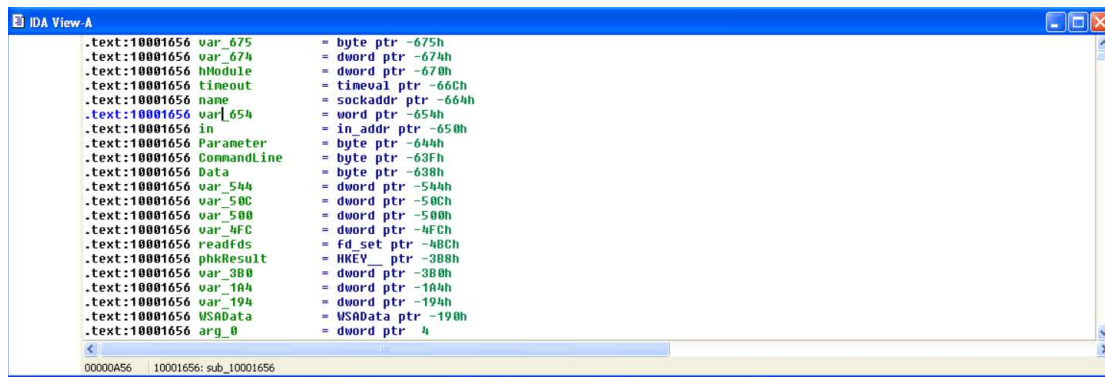
2. Dalla scheda «imports» individuare la funzione «gethostbyname». Qual è l'indirizzo dell'import? Cosa fa la funzione?

Address	Ordinal	Name	Library
10016274		fopen	MSVCRT
100162E4		fprintf	MSVCRT
10016234		fread	MSVCRT
100162DC		free	MSVCRT
100162D8		fseek	MSVCRT
10016278		ftell	MSVCRT
10016240		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
1001624C		memcmp	MSVCRT
100162C8		memcpy	MSVCRT
100162D4		memset	MSVCRT
10016388		mouse_event	USER32

Aprendo la scheda <imports> su IDA, filtrando per nome e premendo la 'g' possiamo vedere che spunta la funzione che stiamo cercando e l'indirizzo è 100163CC.

La funzione 'gethostbyname' è una funzione di sistema utilizzata per ottenere l'indirizzo IP associato a un nome host.

3. Quante sono le variabili locali della funzione alla locazione di memoria 0x10001656?



Nello screen sopra possiamo vedere le variabili locali della funzione alla locazione di memoria 0x10001656, possiamo notare che in totale sono 20 dato che le variabili hanno sempre offset negativo.

4. Quanti sono, invece, i parametri della funzione sopra?

```
.text:10001656 arg_0      = dword ptr 4
.text:10001656
* .text:10001656      sub     esp, 678h
* .text:1000165C      push    ebx
* .text:1000165D      push    ebp
* .text:1000165E      push    esi
* .text:1000165F      push    edi
* .text:10001660      call    sub_10001000
```

A differenza delle variabili, che hanno sempre offset negativo, i parametri hanno offset positivo quindi possiamo vedere che ne abbiamo solo uno chiamato arg_0.

5. Inserire altre considerzioni macro livello sul malware (comportamento)

Utilizzando CFF Explorer ho analizzato il file e ho ricavato il codice hash MD5 e poi l'ho scansionato sul virus total

Malware Analysis_Final (Istantanea Backup3) [In esecuzione] - Oracle VM VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

CFF Explorer VIII - [Malware_U3_W3_L2.dll]

File Settings ?

Malware_U3_W3_L2.dll

Property	Value
File Name	C:\Documents and Settings\Administrator\Desktop\Esercizio_Pratico_U3_W...
File Type	Portable Executable 32
File Info	No match found.
File Size	130.94 KB (134085 bytes)
PE Size	131.00 KB (134144 bytes)
Created	Tuesday 16 August 2022, 13.37.31
Modified	Tuesday 10 May 2011, 15.42.00
Accessed	Tuesday 05 December 2023, 11.58.01
MD5	1A9FD80174AAFECD9A52FD908CB82637
SHA-1	FBE285B8B7FE710724EA35D15948969A709ED33B

Property	Value
CompanyName	Microsoft Corporation
FileDescription	File Encryption Utility
FileVersion	5.1.2201.1329
InternalName	X-doorc
LegalCopyright	(C) Microsoft Corporation. All rights reserved.
OriginalFilename	
ProductName	Microsoft(R) Windows(R) Operating System

File: Malware_U3_W3_L2.dll

- Dos Header
- Nt Headers
 - File Header
 - Optional Header
 - Data Directories [x]
- Section Headers [x]
- Export Directory
- Import Directory
- Resource Directory
- Relocation Directory
- Address Converter
- Dependency Walker
- Hex Editor
- Identifier
- Import Adder
- Quick Disassembler
- Rebuilder
- Resource Editor
- UPX Utility

eb1077b0d96bc7cc19c38b76342113a09666aad47518f1a7536eebf8baad4a

59 / 71

59 security vendors and no sandboxes flagged this file as malicious

eb1077b0d96bc7cc19c38b76342113a09666aad47518f1a7536eebf8baad4a

X-doorc

Size 130.94 KB Last Analysis Date 18 hours ago

peid: comgt: armadillo: overlay:

Community Score

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 19

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

Popular threat label: trojan:ldcafr06cc0df321 Threat categories: trojan Family labels: ldcafr: r06cc0df321

Security vendors' analysis

Vendor	Detection	Vendor	Detection
AhnLab-V3	Backdoor:Win32.Agent.RP408	Alibaba	Backdoor:Win32/ldcafr.9f3a5556
ALYac	Backdoor:Win32	Antiy-AVL	Trojan(Backdoor)/Win32.Agent
Avast	Backdoor:Win32	Avast	Win32.Agent-OLH [Tij]
AVG	Win32.Agent-OLH [Tij]	Avira (no cloud)	BDS/Agent.twe.134160
BitDefender	Backdoor:Win32	Bkav Pro	W32.AIDetect/Malware
ClamAV	Win.Trojan.ldcafr-9937585-0	CrowdStrike Falcon	Win/malicious_confidence_100% (W)
Cylance	Unsafe	Cynet	Malicious (score: 100)
DeepInstinct	MALICIOUS	DrWeb	BackDoor.Siggen.47995

Come possiamo vedere dai risultati si tratta di una backdoor.

Anche analizzandolo con IDA possiamo notare come sia una backdoor questo file.

```

IDA View-A
* 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_100042D0+F2fo
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_100042D0+B5fo
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+198fo
* xdoors_d:10093DE2 align 4
* xdoors_d:10093DE4 ; char aErro[]
xdoors_d:10093DE4 aErro db '-erro',0 ; DATA XREF: sub_10004738+187fo
* xdoors_d:10093DEA align 4
* xdoors_d:10093DEC ; char aStop[]
xdoors_d:10093DEC aStop db '-stop',0 ; DATA XREF: sub_10004738+176fo
0001C174 10093D74: xdoors_d:aBackdoorServer

```