

Cyber Security & Ethical Hacking

GIORNO 2 –MALWARE ANALYSIS

Alejandro Cristino
S11-L2

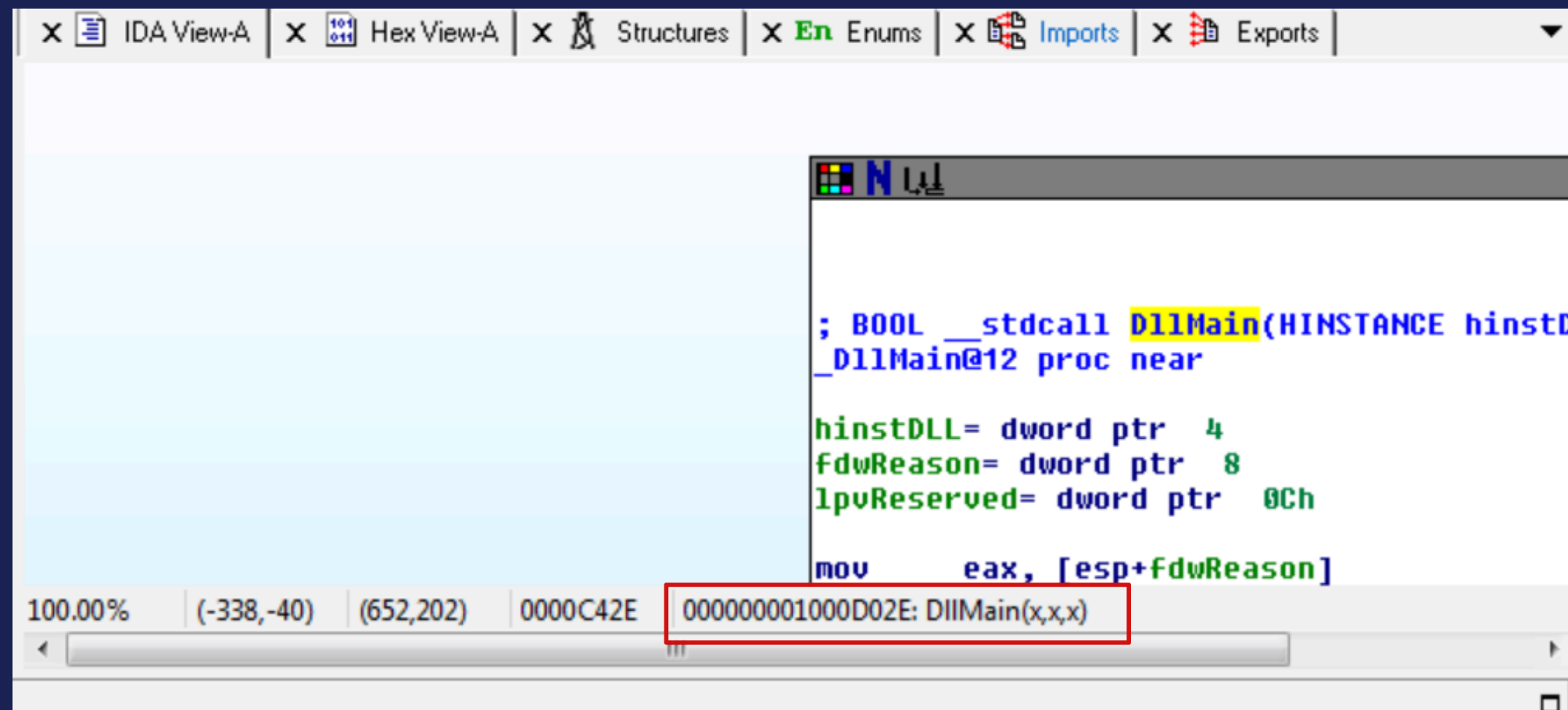
TRACCIA

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 macro livello sul malware (comportamento)

1. Individuare l'indirizzo della DLLMain

Il software IDA carica l'eseguibile permettendo la visualizzazione grafica del programma, tramite la funzione di ricerca è stato possibile recuperare la funzione DLLMain che si trova nell'indirizzo 100D022E



The screenshot shows the IDA Pro interface with the 'Imports' window open. The function 'DLLMain' is listed with the address '00000000100D022E'. The main window displays the assembly code for this function, which is a standard Windows DLL entry point. The code includes parameter declarations for 'hinstDLL', 'fdwReason', and 'lpvReserved', followed by a 'mov' instruction to set 'eax' to '[esp+fdwReason]'. The address '00000000100D022E' is highlighted in the bottom status bar.

```
; BOOL __stdcall DLLMain(HINSTANCE hinstDLL,
_DllMain@12 proc near

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

mov     eax, [esp+fdwReason]
```

100.00% (-338,-40) (652,202) 0000C42E 00000000100D022E: DLLMain(x,x,x)


Indirizzo DLLMain

2.Dalla scheda «imports» individuare la funzione «gethostbyname ».

Qual è l'indirizzo dell'import? Cosa fa la funzione?

Gethostbyname è una funzione di libreria utilizzata per ottenere informazioni sul nome host associato a un determinato indirizzo IP. Questa funzione è considerata obsoleta nelle versioni più recenti di alcune librerie e linguaggi di programmazione, ed è stata sostituita da funzioni più moderne come getaddrinfo.

Indirizzo funzione Gethostbyname



Address	Ordinal	Name	Library
00000000100163BC		waveInStart	WINMM
00000000100163C4	18	select	WS2_32
00000000100163C8	11	inet_addr	WS2_32
00000000100163CC	52	gethostbyname	WS2_32
00000000100163D0	12	inet_ntoa	WS2_32
00000000100163D4	16	recv	WS2_32
00000000100163D8	19	send	WS2_32
00000000100163DC	4	connect	WS2_32
00000000100163E0	15	ntohs	WS2_32
00000000100163E4	9	htons	WS2_32
00000000100163E8	21	setsockopt	WS2_32
00000000100163EC	116	WSACleanup	WS2_32
00000000100163F0	115	WSAStartup	WS2_32

Line 235 of 253

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

Nella versione testuale è emerso che le variabili locali con offset negativo sono 23.

Variabili locali con offset negativo

Locazione di memoria


```
.text:10001656  
.text:10001656 var_675  
.text:10001656 var_674  
.text:10001656 hLibModule  
.text:10001656 timeout  
.text:10001656 name  
.text:10001656 var_654  
.text:10001656 Dst  
.text:10001656 Parameter  
.text:10001656 var_640  
.text:10001656 CommandLine  
.text:10001656 Source  
.text:10001656 Data  
.text:10001656 var_637  
.text:10001656 var_544  
.text:10001656 var_50C  
.text:10001656 var_500  
.text:10001656 Buf2  
.text:10001656 readfds  
.text:10001656 phkResult  
.text:10001656 var_380  
.text:10001656 var_1A4  
.text:10001656 var_194  
.text:10001656 WSAData  
.text:10001656 arg 0
```

```
= byte ptr -675h  
= dword ptr -674h  
= dword ptr -670h  
= timeval ptr -66Ch  
= sockaddr ptr -664h  
= word ptr -654h  
= dword ptr -650h  
= byte ptr -644h  
= byte ptr -640h  
= byte ptr -63Fh  
= byte ptr -63Dh  
= byte ptr -638h  
= byte ptr -637h  
= dword ptr -544h  
= dword ptr -50Ch  
= dword ptr -500h  
= byte ptr -4FCh  
= fd_set ptr -4BCh  
= byte ptr -3B8h  
= dword ptr -3B0h  
= dword ptr -1A4h  
= dword ptr -194h  
= WSAData ptr -190h  
= dword ptr 4
```

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

Nella versione testuale è emerso che i parametri con offset positivo sono in totale 1.

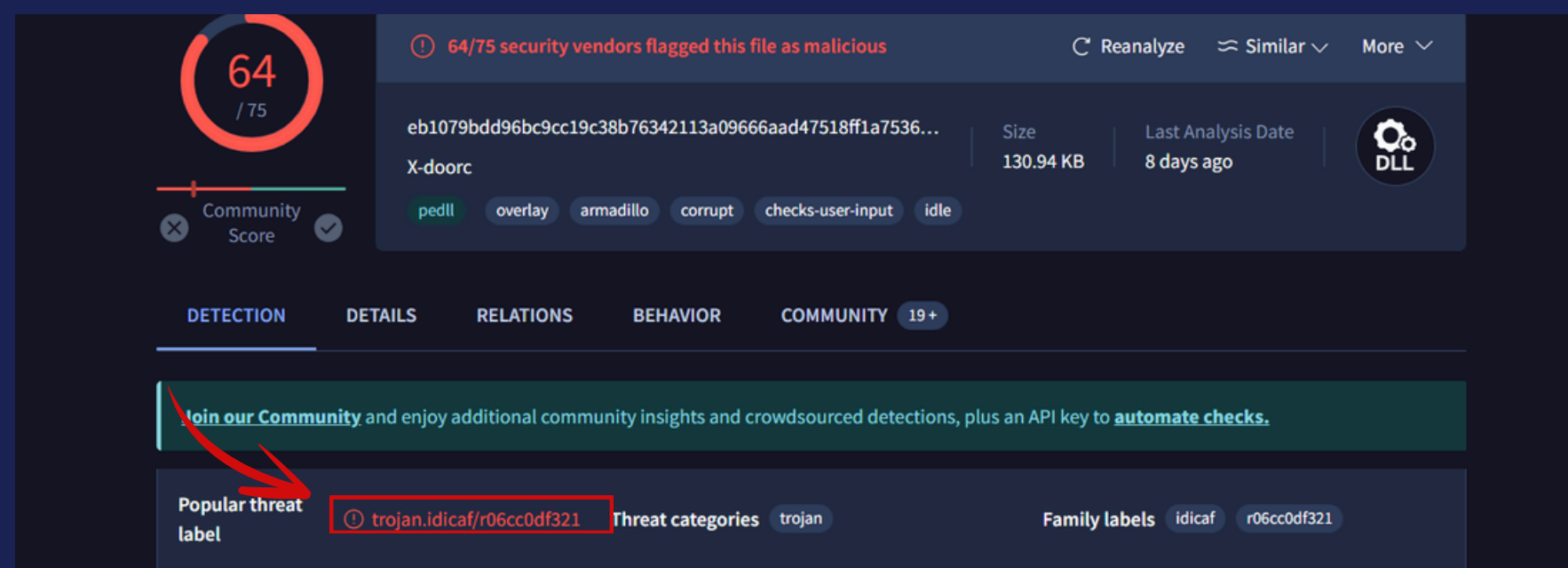
Parametri con offset positivo



```
.text:10001656 Buf2          = byte ptr -4FCh
.text:10001656 readfds       = fd_set ptr -4BCh
.text:10001656 phkResult    = byte 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
.text:10001656             sub     esp, 678h
.text:10001656             push    ebx
```

5. Inserire altre considerazioni macro livello sul malware

Dopo aver effettuato l'analisi del malware tramite IDA, l'individuazione della funzione "gethostbyname" potrebbe suggerire che il malware stia tentando di comunicare con un server remoto utilizzando un nome di dominio piuttosto che un indirizzo IP statico. Questo comportamento può essere indicativo di un tentativo del malware di stabilire una connessione con un server remoto per finalità come l'esfiltrazione di dati, la ricezione di comandi o altre attività malevole legate alla comunicazione in rete. Tale scenario potrebbe far pensare alla presenza di una backdoor, progettata per consentire l'accesso remoto non autorizzato al sistema compromesso. Per ottenere ulteriori informazioni del malware in esame, è stato utilizzato lo strumento VirusTotal, che ci ha fornito informazioni dettagliate in modo più comprensibile rispetto al codice Assembly x86. Come illustrato nell'immagine successiva, il malware si presenta come un trojan che implementa una backdoor nel sistema.



 Cybersecurity Specialist

Cyber Security & Ethical Hacking

GRAZIE.