MALWARE ANALYSIS

ANALISI STATICA AVANZATA CON IDA

RAFAEL MANGO

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.

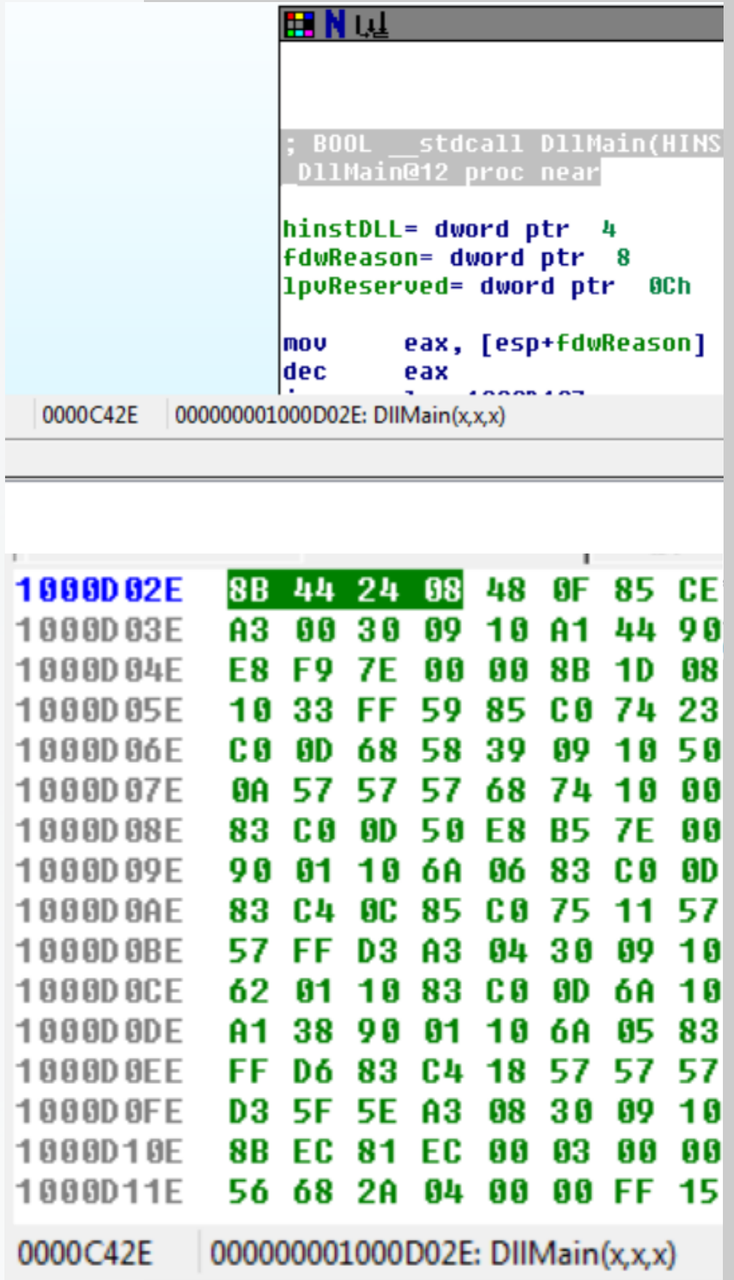
1. INDIVIDUARE L'INDIRIZZODELLA FUNZIONE DLLMAIN(COSÌ COM'È, IN ESADECIMALE)
2. DALLA SCHEDA «IMPORTS» INDIVIDUARE LA FUNZIONE «GETHOSTBYNAME». QUAL È L'INDIRIZZO DELL'IMPORT? COSA FA LA FUNZIONE?
3. QUANTE SONO LE VARIABILI LOCALI DELLA FUNZIONEALLA LOCAZIONE DI MEMORIA 0X10001656?
4. QUANTI SONO, INVECE, I PARAMETRI DELLA FUNZIONE SOPRA?
5. INSERIRE ALTRE CONSIDERAZIONI MACRO LIVELLO SUL MALWARE (COMPORTAMENTO)

1

Individuare l’indirizzodella funzione DLLMain(così com’è, in esadecimale)

come possiamo vedere dall’immagine l’indirizzo della funzione DLLMain e 1OOODO2E

per andare a trovare l’indirizzo siamo andati ad aprire il malware con IDA , come nell’immagine in alto a destra siamo andati a evidenziare le prime righe di codice e sotto il codice vicino alla riga con scritto OOOOC42E troviamo l’indirizzo della funzione DLLMain per verificare che non ci sbagliamo andiamo nella finestra Hex View-a e ci ritroveremo nella finestra come nell’immagine in basso a destra



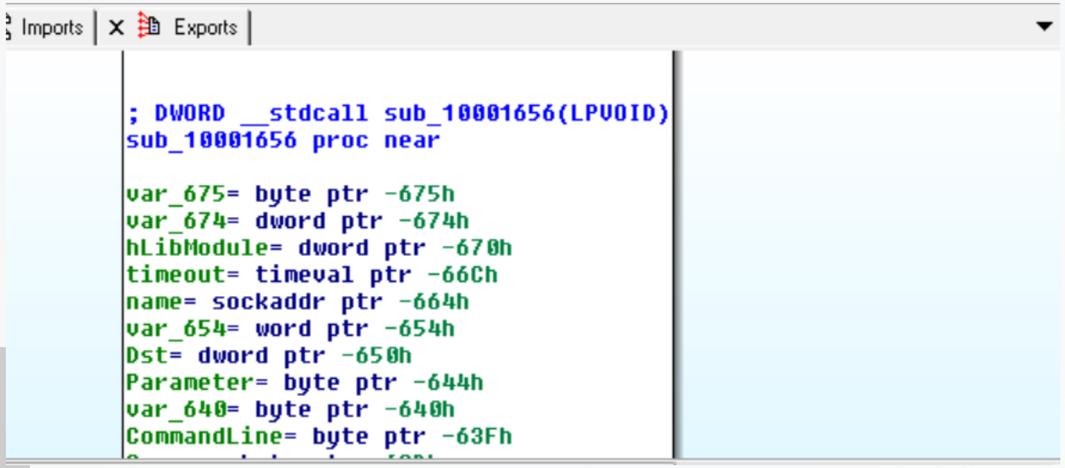
2

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

Per andare a trovare la funzione gethostbyname siamo andati nella scheda imports come in figura cliccandoci ci troveremo nella scheda in figura. tornando nella scheda IDA ViewA troviamo anche l'indirizzo che sarà 10001656 .
La funzione gethostbyname è una funzione di libreria standard di Windows utilizzata per ottenere informazioni sulle entità host, come gli indirizzi IP, utilizzando il nome host.
Questo tipo di funzione è comunemente utilizzato nelle applicazioni di rete per convertire nomi di dominio in indirizzi IP, consentendo alle applicazioni di comunicare con gli host identificati da nomi simbolici anziché da indirizzi numerici.

Address	Ordinal	Name	Library
000000...		waveInPrepareHeader	WINMM
000000...		waveInAddBuffer	WINMM
000000...		waveInStart	WINMM
000000...	18	select	WS2_32
000000...	11	inet_addr	WS2_32
000000...	52	gethostbyname	WS2_32
000000...	12	inet_ntoa	WS2_32
000000...	16	recv	WS2_32
000000...	19	send	WS2_32
000000...	4	connect	WS2_32
000000...	15	ntohs	WS2_32
000000...	9	htons	WS2_32
000000...	21	setsockopt	WS2_32

Line 235 of 253



3 Quante sono le variabili locali della funzionealla locazione di memoria 0x10001656?

possiamo vedere che ci sono diverse variabili locali definite all'interno della funzione sub_10001656. Queste variabili locali sono elencate con i loro relativi offset rispetto allo stack frame, indicati dai valori negativi (ad esempio, -675h, -674h, -670h, ecc.).

Contando il numero di variabili locali definite nel codice fornito, possiamo determinare quanti sono:

1. var_675
2. var_674
3. hLibModule
4. timeout
5. name
6. var_654
7. Dst
8. Parameter
9. var_640
10. CommandLine
11. Source
12. Data
13. var_637
14. var_544
15. var_50C
16. var_500
17. Buf2
18. readfds
19. phkResult
20. var_3B0
21. var_1A4
22. var_194
23. WSAData

Quindi, ci sono 23 variabili locali definite all'interno della funzione sub_10001656 alla locazione di memoria 0x10001656.

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

La funzione sub_10001656 accetta un solo parametro, come indicato dalla dichiarazione della procedura sub_10001656:

sub_10001656 proc near

E dalla sua firma:

; DWORD __stdcall sub_10001656(LPVOID)

Il commento sopra la firma indica che la funzione accetta un parametro di tipo LPVOID, che è un puntatore a void. Inoltre, l'etichetta arg_0 definita come dword ptr 4 indica il parametro passato alla funzione attraverso il registro eax.

Quindi, la funzione sub_10001656 accetta un solo parametro.

5 Inserire altre considerazioni macro livello sul malware

Guardando il malware ho notato che ci sarebbero altre considerazioni macro livello da fare . Di seguito ho riportato alcune considerazioni sul malware, basate sul comportamento e sulle caratteristiche generali che potrebbero essere rilevanti per un'analisi più approfondita:

Furtività: Il malware potrebbe cercare di mascherare la propria presenza, ad esempio mimetizzandosi come un file legittimo o utilizzando tecniche di occultamento per evitare la rilevazione da parte di software antivirus o altri strumenti di sicurezza.
Persistenza: Potrebbe tentare di mantenere la sua presenza nel sistema dopo il riavvio, ad esempio aggiungendo voci al registro di sistema o inserendo codice nei punti di avvio del sistema operativo.
Comunicazione remota: Potrebbe tentare di stabilire una comunicazione con un server remoto per ricevere istruzioni, inviare dati sensibili o scaricare componenti aggiuntivi. Questa comunicazione potrebbe essere criptata o utilizzare protocolli non standard per sfuggire alla rilevazione.
Rubare informazioni: Il malware potrebbe cercare di rubare informazioni sensibili, come credenziali di accesso, dati finanziari o altre informazioni personali memorizzate sul sistema infetto.
Infezione di file: Potrebbe cercare di infettare altri file nel sistema per diffondere ulteriormente il malware e garantire la sua persistenza.
Modifiche del sistema: Potrebbe effettuare modifiche non autorizzate al sistema, ad esempio disabilitando il firewall, bloccando l'accesso a determinati siti web o modificando le impostazioni di sicurezza.
Utilizzo delle risorse di sistema: Il malware potrebbe consumare risorse di sistema, come CPU, memoria o larghezza di banda di rete, riducendo le prestazioni complessive del sistema e causando rallentamenti o blocchi.
Evasione delle tecniche di analisi: Potrebbe essere progettato per eludere l'analisi statica o dinamica utilizzando tecniche come l'opacizzazione del codice, l'auto-modifica o il rilevamento dell'ambiente di analisi.
Polimorfismo: Il malware potrebbe utilizzare tecniche di polimorfismo per modificare la propria struttura e il proprio comportamento nel tempo al fine di eludere la rilevazione da parte dei motori antivirus.
Attacchi mirati: In alcuni casi, il malware potrebbe essere progettato per condurre attacchi mirati contro specifici obiettivi, come aziende o istituzioni governative, al fine di rubare informazioni sensibili o compromettere la loro sicurezza.
È importante esaminare attentamente il contesto e condurre un'analisi approfondita per comprendere appieno il comportamento e le intenzioni del codice sospetto.

THANK YOU