Report S11-L2
Analisi Statica Avanzata con Ida Pro

Sommario

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
2. Dalla scheda «imports» individuare la funzione «gethostbyname». Qual è l'indirizzo dell'import?
3. Quante sono le variabili locali della funzione alla locazione di memoria 0x10001656?
4. Quanti sono, invece, i parametri della funzione sopra?

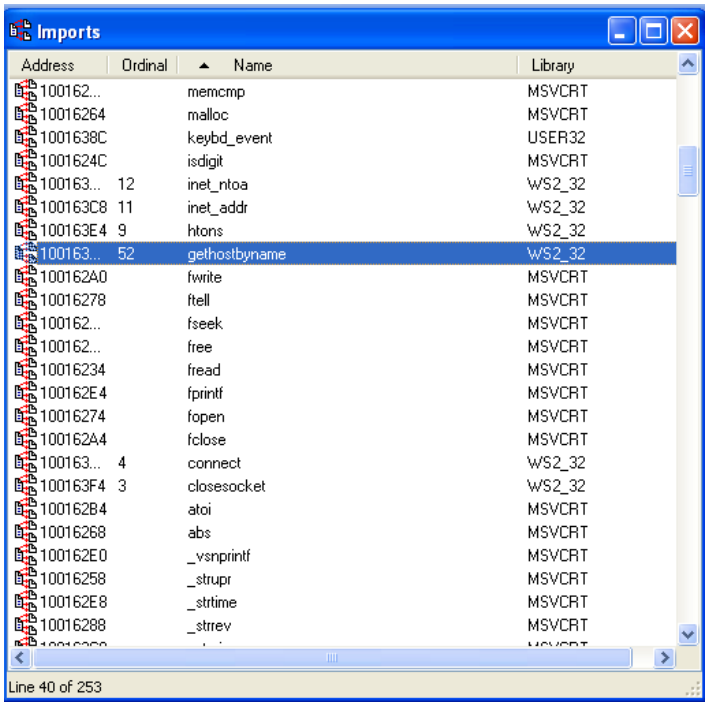
1°task – Individuazione indirizzo della funzione DLLMain

La funzione DllMain è una funzione speciale nelle **Dynamic Link Library** (**DLL**) in ambienti Windows che viene chiamata automaticamente quando una DLL viene caricata o scaricata da un processo, o quando si verificano determinati eventi legati alla DLL.
Questa funzione può essere utilizzata per l'inizializzazione e la pulizia delle risorse della DDL, per l'esecuzione di operazioni specifiche del thread e altro ancora, a seconda del motivo per cui viene chiamata.
Questo fornisce un punto di ingresso per l'inizializzazione e la pulizia della DLL.

```
.text:1000D02E ; ||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||||||||||||||
ogram control flow)02E
.text:1000D02E
.text:1000D02E ; BOOL __stdcall DllMain(HINSTANCE hinstDLL,DWORD fdwReason,LPVOID lpvReserved)
.text:1000D02E _DllMain@12     proc near               ; CODE XREF: DllEntryPoint+4B↓p
.text:1000D02E                                         ; DATA XREF: sub_100110FF+2D↓o
```

Nel contesto dell'analisi statica avanzata di un file PE con IDA Pro, l'identificazione della funzione **DllMain** all'indirizzo **0x1000D02E** potrebbe essere un punto chiave per comprendere il comportamento del malware in quanto rappresenta il punto di ingresso esatto durante il caricamento della DLL e costituisce il luogo in cui il malware avvia le sue attività iniziali.
Da qui, il malware può propagarsi, eseguire codice dannoso e intraprendere altre azioni dannose
L'analisi di questa istruzione fornisce un quadro iniziale fondamentale delle attività del malware, consentendo agli analisti di identificare le minacce e sviluppare strategie di mitigazione adeguate.

2°task- Individuazione funzione gethostbyname e indirizzo dell'import



Con Ida Pro è stato possibile individuare la funzione "**gethostbyname**", componente fondamentale della Libreria importata WS2_32, all'indirizzo di Import **0x100163CC.**

La **funzione gethostbyname** è una funzione di libreria standard di Windows inclusa nella libreria WS2_32 (**Winsock 2**).
Questa funzione è comunemente ***utilizzata per risolvere il nome host in un indirizzo IP***
WS2_32 è la libreria di Winsock 2, che fornisce API per la programmazione di socket su piattaforme Windows.
È ampiamente utilizzata per ***implementare la comunicazione di rete*** e, nel contesto del malware, l'utilizzo di WS2_32 suggerisce che il codice del malware sta coinvolto in operazioni di rete avanzate, quali connessioni di rete, invio o ricezione di dati attraverso socket.

L'uso della funzione **gethostbyname** in un malware suggerisce la possibilità che il programma stia cercando di stabilire una ***connessione con un server remoto***. Questo comportamento è tipico dei malware che comunicano con un server di comando e controllo (C2), un'infrastruttura centrale che guida le azioni del malware. Attraverso questa connessione, il malware potrebbe ricevere istruzioni da operatori malintenzionati, inviare dati compromettenti o persino scaricare componenti aggiuntivi.

Un aspetto notevole è che l'uso di **gethostbyname** potrebbe essere ***una tattica per nascondere l'indirizzo IP del server remoto***. Invece di codificare staticamente l'indirizzo IP nel codice, il malware risolve dinamicamente il nome host durante l'esecuzione. Questa strategia rende più difficile per i ricercatori di sicurezza e gli strumenti di analisi tracciare il server remoto, poiché l'indirizzo IP potrebbe variare dinamicamente nel tempo o essere distribuito su diversi nomi host.

In sintesi, l'utilizzo di **gethostbyname** in un contesto malware suggerisce che il programma sta cercando di stabilire una connessione con un server remoto e potrebbe adottare misure per complicare la tracciabilità dell'indirizzo IP del server. Questa dinamicità nel risolvere il nome host al momento dell'esecuzione rende il malware più sfuggente e difficile da individuare per i ricercatori di sicurezza.

3° e 4° task – Numero variabili locali e parametri della funzione alla locazione di memoria 0x10001656

```
.text:10001656 ; ||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||||||||||||||
.text:10001656
.text:10001656
.text:10001656 ; DWORD __stdcall sub_10001656(LPVOID)
.text:10001656 sub_10001656    proc near               ; DATA XREF: DllMain(x,x,x)+C8↓o
.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
ontrol flow)0001656 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
```

Tramite disassemblaggio operato sempre con Ida pro sono state individuate **20** variabili locali ed un solo parametro relativi alla funzione rintracciabile all'indirizzo 0x10001656.

La presenza di un **numero significativo di variabili locali** indica una potenziale complessità del malware che potrebbe essere coinvolto in operazioni avanzate, come manipolazioni di dati o l'esecuzione di processi sofisticati.
Inoltre, potrebbe trattarsi di una tattica per rendere il codice più difficile da interpretare, mascherando le intenzioni del malware e complicando l'analisi statica.

Il **parametro di funzione arg_0**, fungendo da argomento di input, potrebbe svolgere un ruolo cruciale nella trasmissione di informazioni cruciali alla funzione durante l'esecuzione del malware.
La sua presenza suggerisce la possibilità che arg_0 sia coinvolto nel riferimento a strutture dati esterne, indicando una connessione diretta con dati provenienti da fonti esterne al contesto della funzione.
Inoltre, arg_0 potrebbe svolgere un ruolo nel coordinamento con altri componenti del malware, partecipando a un processo più ampio all'interno del quale le informazioni vengono scambiate o elaborate in modo collaborativo.
Questo coordinamento tra i componenti potrebbe indicare un'architettura più sofisticata e la presenza di funzionalità avanzate all'interno del malware.